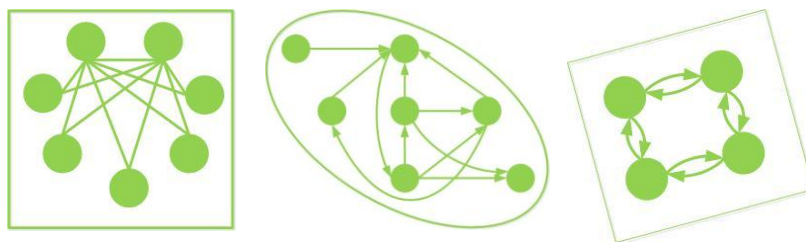


Э.Р. Зарипова, М.Г. Кокотчикова



**Дискретная математика
Часть III
Теория графов**

**Москва
Российский университет дружбы
народов
2013**

Э.Р. Зарипова, М.Г. Кокотчикова

**Дискретная математика
Часть III
Теория графов**

**Москва
Российский университет дружбы
народов
2013**

ББК 22.12

Г 17

Утверждено
РИС Ученого совета
Российского университета
дружбы народов

Рецензент –

Доктор физико-математических наук, профессор Севастьянов Л.А.

Зарипова Э.Р., Кокотчикова М.Г.

С 17 Дискретная математика. Часть III. Теория графов:
Учеб. пособие. – М.: Изд-во РУДН, 2013. – с.

ISBN 5-209-01493-2

В пособии излагаются основы теории графов и алгоритмов на графах. Книга является продолжением курса дискретной математики: «Часть I. Комбинаторика» и «Часть II. Математическая логика».

Подготовлено на кафедре «Системы телекоммуникаций». Предназначено для студентов I, II курсов математических и компьютерных специальностей высших учебных заведений.

ISBN 5-209-01493-2

ББК 22.12

©Издательство Российского университета дружбы народов, 2013
Э.Р. Зарипова, М.Г. Кокотчикова, 2013

I. КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ

Тема 1. Графы. Неориентированные графы: основные понятия; маршруты, цепи, циклы; связность; деревья и леса

Основные понятия

Пусть V – непустое множество, $V^{(2)}$ – множество всех его двухэлементных подмножеств, т.е. $(V_1, V_2) \in V^{(2)}$, если $V_1, V_2 \in V$.¹

Определение. Неориентированным графом (или просто графом) называется пара $G = (V, E)$, где $E \subseteq V^{(2)}$.

Элементы множества V называются вершинами графа, а элементы множества E – ребрами.² Далее будем рассматривать граф $G = (V, E)$.

Графы удобно представлять в виде рисунков, состоящих из точек, изображающих вершины, и линий, соединяющих некоторые из вершин, и изображающих ребра.

Пример 1. Рассмотрим следующий граф (Рис. 1):

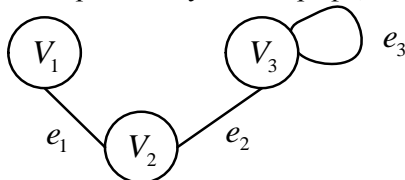


Рис. 1. Исходный граф.

Множество вершин: $V = \{V_1, V_2, V_3\}$, множество ребер: $E = \{e_1, e_2, e_3\}$, где $e_1 = (V_1, V_2)$, $e_2 = (V_2, V_3)$ и $e_3 = (V_3, V_3)$.

Определение. Если ребро e соединяет вершины V_1 и V_2 ,

¹ V от слова *vertex*, в переводе с английского языка *вершина*.

² E от слова *edge*, в переводе с английского языка *ребро*.

$e = (V_1, V_2) \in \mathbf{E}$, то вершины V_1, V_2 называются смежными, а ребро $e = (V_1, V_2)$ называется ребром, инцидентным вершинам V_1 и V_2 .

Определение. Если \mathbf{V} и \mathbf{E} – конечные множества, то G называется конечным графом.

Определение. Если ребро e инцидентно вершинам V_1 и V_2 , то такие вершины называются граничными точками ребра e .

Определение. Если вершины V_1 и V_2 – граничные точки ребра e и вершина V_1 совпадает с вершиной V_2 , то ребро e называется петлей. (В этом случае вершина V_1 смежна сама с собой.)

Примером петли является ребро $e_3 = (V_3, V_3)$ из примера 1.1.

Определение. Если вершины V_1 и V_2 одновременно инцидентны ребрам e_1 и e_2 , то ребра e_1 и e_2 – называются параллельными ребрами.

Определение. Ребра e_1 и e_2 называются смежными, если они имеют хотя бы одну общую граничную точку.

Замечание. *Смежность* является отношением между двумя подобными элементами (между вершинами или между ребрами), а *инцидентность* есть отношение между *разнородными* элементами.

Пример 2. Рассмотрим следующий граф (Рис. 2):

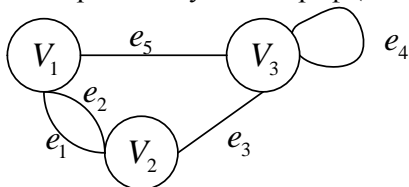


Рис. 2. Исходный граф.

В данном графе существуют параллельные ребра e_1 и e_2 , ребра e_3 и e_5 являются смежными, а ребра e_1 и e_4 не смежные, также как и e_2 и e_4 . Петля – e_4 .

Определение. Число ребер, инцидентных вершине V (петля учитывается дважды), называется степенью вершины V и обозначается $\delta(V)$ ³. Вершина V изолирована, если $\delta(V)=0$. В случае петли, например, в вершине V_1 , $\delta(V_1)=2$.

Пример 3. Найти степени вершин для графа. Рассмотрим следующий граф (Рис. 3):

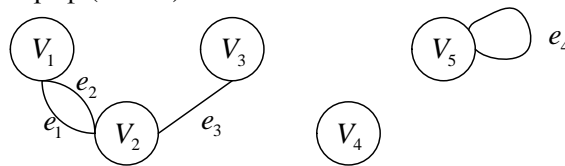


Рис. 3. Исходный граф.

$$\delta(V_1)=2, \delta(V_2)=3, \delta(V_3)=1, \delta(V_4)=0, \delta(V_5)=2.$$

Определение. Граф называется вырожденным (пустым), если все его вершины являются изолированными.

Пример 4. Пример изолированного графа представлен на Рис. 4.

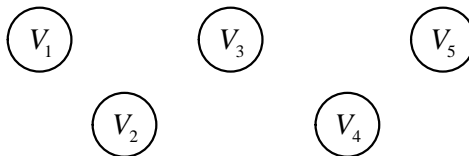


Рис. 4. Изолированный граф.

Рассмотрим два графа $G=(\mathbf{V},\mathbf{E})$ и $G^*=(\mathbf{V}^*,\mathbf{E}^*)$, и пусть существует биекция (взаимно однозначное отображение) – $\varphi: \mathbf{V} \rightarrow \mathbf{V}^*$.

Определение. Если для любых вершин V_i и V_j графа G их

³ δ – греческая буква дельта.

образы $\varphi(V_i)$ и $\varphi(V_j)$ смежны в G^* тогда и только тогда, когда вершины V_i и V_j смежны в G , то эта биекция называется изоморфизмом графа G на граф G^* . Если такой изоморфизм существует, то граф G *изоморфен* графу G^* .

Пример 5. Пример изоморфных графов представлен на Рис. 5.

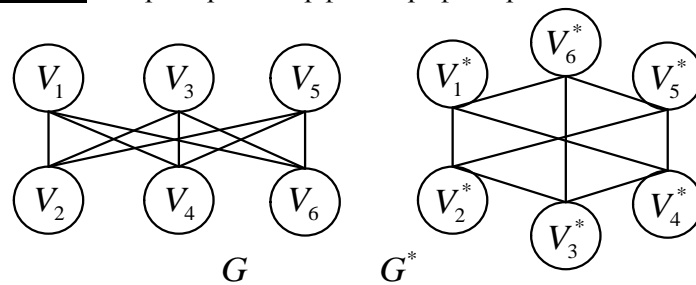


Рис. 5. Изоморфные графы.

Для графов G и G^* вершины пронумерованы соответствующим образом.

Пример 6. Пример неизоморфных графов представлен на Рис. 6.

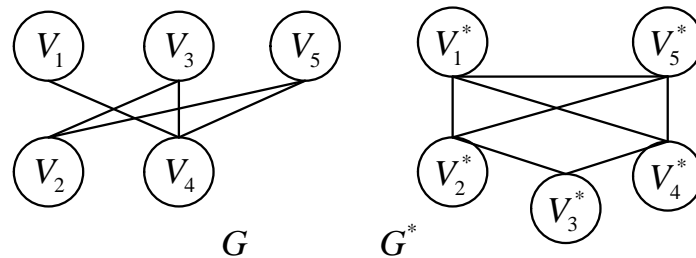


Рис. 6. Неизоморфные графы.

Для этих двух графов G и G^* число вершин одинаковое, но, например, степень вершины V_1 графа G равна единице, а в графе G^* нет вершин, степень которых меньше двух. Читателю предлагается найти другие признаки неизоморфности.

Пример 7. Даны два графа G и G^* (Рис. 7). Являются ли эти два графа изоморфными? Если графы являются изоморфными, найдите биекцию.

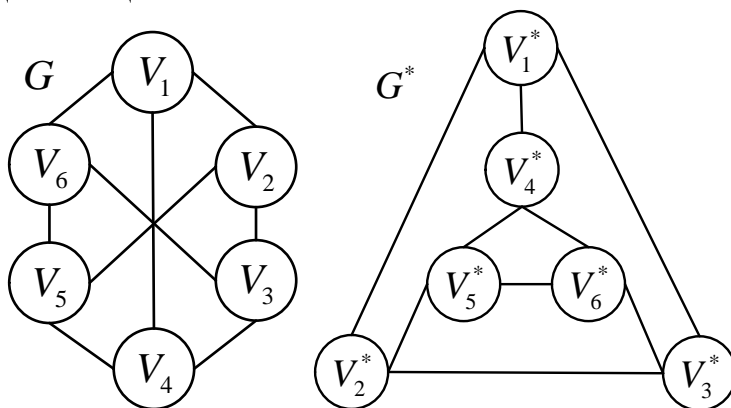


Рис. 7. Исходные графы.

Теорема о числе вершин нечетной степени в графе. В конечном графе число вершин нечетной степени четно.

Доказательство. Нечетная степень вершины означает, что число ребер, инцидентных данной вершине, нечетно. Докажем, что количество таких вершин в графе четно.

Рассмотрим конечный граф $G = (\mathbf{V}, \mathbf{E})$. Пусть $|\mathbf{V}|^4$ и $|\mathbf{E}|$ – число вершин и ребер соответственно.

Утверждение. $\sum_{V \in \mathbf{V}} \delta(V) = 2 \cdot |\mathbf{E}|.$

(Так как каждое новое ребро добавляет по единице к степеням двух вершин или двойку в случае петли к одной вершине.)

Разобьем множество вершин \mathbf{V} на два множества: $\mathbf{V} = \mathbf{V}_2 + \mathbf{V}_1.$

⁴ $|M|$ означает мощность множества, количество элементов множества.

1) V_2 – множество всех вершин, имеющих четные степени,
 $\sum_{V \in V_2} \delta(V)$ – сумма степеней вершин четной степени.

2) V_1 – множество всех вершин, имеющих нечетные степени,
 $\sum_{V \in V_1} \delta(V)$ – сумма степеней вершин нечетной степени.

$$\begin{aligned} \sum_{V \in V} \delta(V) &= \sum_{V \in V_2} \delta(V) + \sum_{V \in V_1} \delta(V) \Rightarrow \\ \sum_{V \in V_1} \delta(V) &= \sum_{V \in V} \delta(V) - \sum_{V \in V_2} \delta(V). \end{aligned} \quad (1)$$

Так как по утверждению $\sum_{V \in V} \delta(V) = 2 \cdot |E|$, то эта сумма является четной. Сумма $\sum_{V \in V_2} \delta(V)$ также четна, так как суммируются четные степени вершин.

$\Rightarrow \sum_{V \in V_1} \delta(V)$ – эта сумма также четна, так как является конечной суммой четных чисел.

В сумме $\sum_{V \in V_1} \delta(V)$ суммируются нечетные числа, формула (1).

Для того, чтобы сумма $\sum_{V \in V_1} \delta(V)$ была четной, количество слагаемых должно быть четно. \Rightarrow Количество вершин нечетной степени четно. ■⁵

Определение. Граф (без петель) называется полным, если любые две его вершины смежны. Полный граф с n вершинами имеет C_n^2 ребер.

Замечание. Из комбинаторики, C_n^k – число сочетаний из n по k , которое определяется по следующей формуле:

⁵ ■ – здесь и далее символ конца доказательства.

$C_n^k = \frac{n!}{(n-k)!k!} = \frac{1 \cdot 2 \cdot \dots \cdot n}{1 \cdot 2 \cdot \dots \cdot (n-k) \cdot 1 \cdot 2 \cdot \dots \cdot k}$. Тогда C_n^2 – число

сочетаний из n по 2, вычислим это число:

$$C_n^2 = \frac{n!}{(n-2)!2!} = \frac{1 \cdot 2 \cdot \dots \cdot (n-3) \cdot (n-2) \cdot (n-1) \cdot n}{1 \cdot 2 \cdot \dots \cdot (n-3) \cdot (n-2) \cdot 1 \cdot 2} = \frac{(n-1) \cdot n}{2}.$$

Получаем, что $C_n^2 = \frac{(n-1) \cdot n}{2}$.

Замечание по комбинаторике. Для числа сочетаний существует два равнозначных обозначения: C_n^k и $\binom{n}{k}$, первое обозначение принято во французской школе, а второе в американской. Обратите внимание на положение индексов, они противоположные. Для C_n^k нижний индекс n обозначает число элементов множества, из которого осуществляется выбор, а верхний индекс k – количество элементов в выборке. В американских обозначениях эти индексы имеют такое же значение, только меняются местами.

Определение. Граф (с петлями) называется полным, если любые две его вершины смежны, а также граф имеет петли в каждой вершине. Такой граф имеет по формуле сочетаний с повторениями $C_n^2 = C_{n+2-1}^2 = C_{n+1}^2 = \frac{(n+1)!}{(n+1-2)!2!} = \frac{(n+1)n}{2}$ ребер.

Замечание по комбинаторике. Число C_n^k называется числом сочетаний с повторениями. Число всех сочетаний с повторениями из n элементов по k вычисляется по формуле:

$$C_n^k = C_{n+k-1}^{n-1} = C_{n+k-1}^k = \frac{(n+k-1)!}{(n+k-1-k)!k!} = \frac{(n+k-1)!}{(n-1)!k!}.$$

Определение. Граф $G_1 = (V_1, E_1)$ называется подграфом графа $G = (V, E)$ при соблюдении следующих условий:

1. $V_1 \subseteq V, E_1 \subseteq E$.

2. Если ребро $e \in \mathbf{E}_1$ инцидентно вершинам V_1 и $V_2 \in \mathbf{V}_1$, то ребро $e \in \mathbf{E}$, также инцидентно вершинам $V_1, V_2 \in \mathbf{V}$.

Маршруты, цепи, циклы

Определение. Конечная последовательность ребер e_1, e_2, \dots, e_n графа $G = (\mathbf{V}, \mathbf{E})$ (не обязательно различных) называется маршрутом длины n , если существует последовательность $V_0, V_1, V_2, \dots, V_n$ вершин (не обязательно различных), таких, что e_i инцидентно вершинам V_{i-1} и V_i , $i = \overline{1, n}$. (Здесь номера вершин и ребер показывают последовательность в маршруте, а не нумерацию в графе.) Маршрут замкнут, если $V_0 = V_n$ (циклический маршрут).

Определение. Маршрут называется цепью, если все его ребра различны и простой цепью, если все его вершины различны (в этом случае и все его ребра различны).

Определение. Замкнутая цепь называется циклом.

Определение. Простая замкнутая цепь называется простым циклом.

Связность

Определение. Граф $G = (\mathbf{V}, \mathbf{E})$ называется связным, если каждая пара различных вершин может быть соединена, по крайней мере, одной цепью. В противном случае граф называется несвязным.

Пример 8. Пример связного и несвязного графов представлен на Рис. 8.

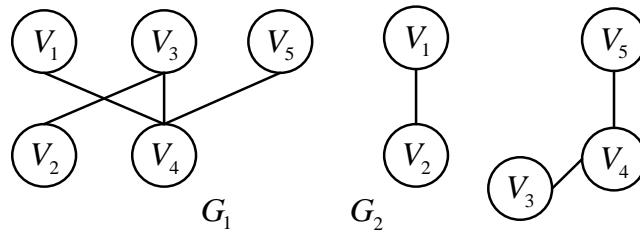


Рис. 8. Связный и несвязные графы.

На Рис. 8 граф G_1 – связный, а граф G_2 – несвязный.

Теорема о связности графа. Граф $G = (V, E)$ связан тогда и только тогда, когда множество его вершин нельзя разбить на два непустых подмножества $V = V_1 + V_2$ так, что обе граничные точки каждого ребра находятся в одном и том же подмножестве.

Доказательство. (От противного в обе стороны.)

1) (слева направо) Пусть $G = (V, E)$ несвязен. Выберем произвольную вершину V_i из множества V_1 . Множество V_1 состоит из вершины V_i и всех вершин, которые могут быть соединены с V_i цепью.

Так как G – несвязен, то $\exists V_2 = V \setminus V_1$ ⁶ и $V_2 \neq \emptyset$. По построению множества V_1 ни одно ребро из множества V_1 не соединяет вершину из множества V_1 ни с одной вершиной из множества V_2 , т.е. \exists разбиение G на непустые подмножества.

2) (справа налево) Пусть \exists разбиение графа G на непустые подмножества V_1 и V_2 . Произвольным образом выберем вершины V_i из V_1 и V_j из V_2 . Цепь, соединяющая вершины V_i и V_j должна содержать минимум одно ребро, содержащее граничные точки в обоих множествах V_1 и V_2 . Так как такого

⁶ Символ \exists означает квантор существования, $\exists x$ - существует x .

ребра не существует, то G – несвязен. ■

Деревья и леса

Определение. Граф называется деревом, если он связан и не имеет циклов. Обозначается буквой T ⁷.

Определение. Граф, не имеющий циклов и состоящий из k компонентов, называется лесом из k деревьев. Обозначается буквой F ⁸.

Пример 9. Пример деревьев (Рис. 9).

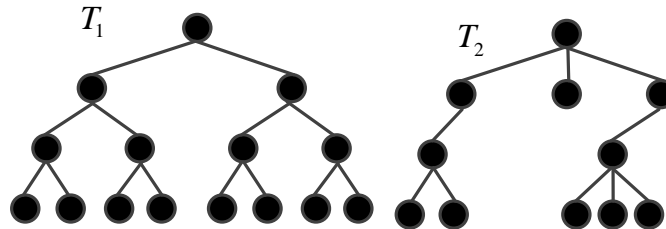


Рис. 9. Деревья.

На Рис. 9 T_1 и T_2 являются деревьями, отметим, что дерево T_1 является полным бинарным деревом, так как из каждой вершины идут по два ребра. Деревья T_1 и T_2 составляют лес F из двух деревьев.

Пример 10. Пример произвольного дерева и графа (Рис. 10).

⁷ T - из английского tree, дерево.

⁸ F - из английского forest, лес.

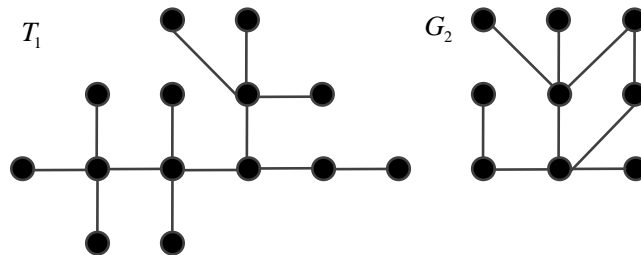


Рис. 10. Дерево и граф.

На Рис. 10 представлены дерево T_1 и граф G_2 . Отметим, что граф G_2 не является деревом, так как в нем есть цикл.

Определение. Если дерево T является подграфом графа G , то ребра графа G , принадлежащие дереву T , называются ветвями дерева T , а ребра, не принадлежащие дереву T , называются хордами относительно дерева T .

Пример 11. Пример ветвей и хорд.

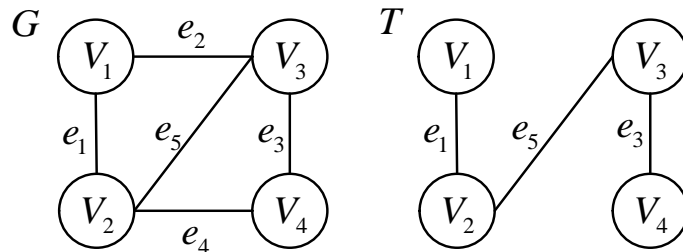


Рис. 11. Граф и дерево.

Из Рис. 11 видно, что дерево T является подграфом графа G . Дерево T получилось путем удаления ребер e_2, e_4 из графа G . Следовательно, ребра e_1, e_3, e_5 графа G являются ветвями дерева T , а ребра e_2, e_4 – хорды относительно дерева T .

Теорема о количестве ребер для дерева с n вершинами.

Доказательство.

Нужно доказать, что количество ребер в дереве не больше $(n-1)$, иначе образуется цикл, и не меньше $(n-1)$, иначе образуется лес.

- 1) Удаление одного ребра разбивает дерево на 2 компоненты связности, то есть превращает его в лес из двух деревьев, граф становится несвязным. Удаление второго ребра превращает дерево в лес из 3 деревьев, и так далее. Удаление $(n-1)$ -го ребра превращает дерево в лес из n деревьев, каждое из которых является изолированной вершиной.
- 2) Добавление любого ребра, после $(n-1)$ образует цикл с ребрами, составляющими дерево.
- 3) Каждое дерево с n вершинами имеет в точности $(n-1)$ ребро. ■

Пример 12. Рассмотрим дерево, состоящее из 7 вершин (Рис. 12). Это дерево содержит 6 ребер. Удаляя по одному из ребер последовательно, получаем сначала лес, состоящий из двух деревьев на Рис. 12, потом лес из трех деревьев на следующем рисунке, и в конечном итоге получаем лес из семи деревьев, каждое из которых является изолированной вершиной.

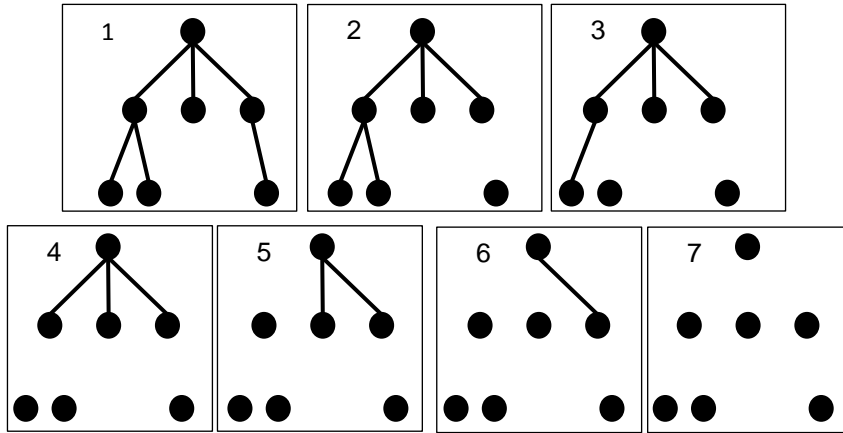


Рис. 12. Пример к теореме о количестве ребер для дерева с семью вершинами.

Тема 2. Ориентированные графы: основные понятия; ориентированные маршруты, пути, контуры; сильная связность. Ориентированные деревья

Основные понятия

Во многих случаях ребрам графа необходимо задать *ориентацию* или *направление*. Отличие орграфов⁹ от неорграфов в том, что у неорграфов граничные точки ребра образуют *неупорядоченную пару*, а в случае орграфа граничные точки дуги образуют *упорядоченную пару*.

Определение. Ориентированным графом (или орграфом) называется пара $G = (\mathbf{V}, \mathbf{E})$, где \mathbf{V} – непустое множество вершин, а $\mathbf{E} \subseteq \mathbf{V}^2 = \mathbf{V} \times \mathbf{V}$ – множество дуг.

Определение. Если дуга $e = \langle V_1, V_2 \rangle \in \mathbf{E}$, то говорят, что вершина V_1 смежна с V_2 , а дуга e отрицательно инцидентна вершине V_2 и положительно инцидентна вершине V_1 .

Пример 13. Рассмотрим 3 ориентированных графа: G_1 , G_2 и G_3 (Рис. 13).

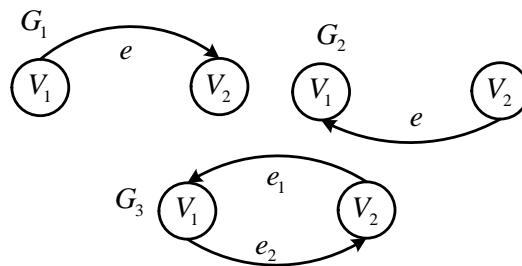


Рис. 13. Ориентированные графы.

⁹ Ориентированный граф или орграф.
Неориентированный граф или неорграф.

Для графа G_1 на Рис. 13 вершина V_1 смежна с вершиной V_2 , дуга e отрицательно инцидентна вершине V_2 и положительно инцидентна вершине V_1 . Для второго графа G_2 , наоборот, вершина V_2 смежна с вершиной V_1 , но вершина V_1 не смежна с вершиной V_2 , дуга e отрицательно инцидентна вершине V_1 и положительно инцидентна вершине V_2 . Для следующего графа G_3 дуга e_1 отрицательно инцидентна вершине V_1 и положительно инцидентна вершине V_2 , а дуга e_2 отрицательно инцидентна вершине V_2 и положительно инцидентна вершине V_1 , при этом обе вершины являются смежными друг с другом.

Определение. Рассмотрим графы $G=(\mathbf{V},\mathbf{E})$ и $G^*=(\mathbf{V}^*,\mathbf{E}^*)$ и пусть \exists биекция $\varphi:\mathbf{V}\rightarrow\mathbf{V}^*$.

Если для любых вершин V_1 и V_2 графа G образ $\varphi(V_1)$ смежен с образом $\varphi(V_2)$ в G^* , тогда и только тогда, когда вершина V_1 смежна вершине V_2 в графе G , то эта биекция называется изоморфизмом графа G на граф G^* . Если такой изоморфизм существует, то граф G изоморфен графу G^* .

Пример 14. Рассмотрим два графа G и G^* (Рис. 1. 1а и 14б).

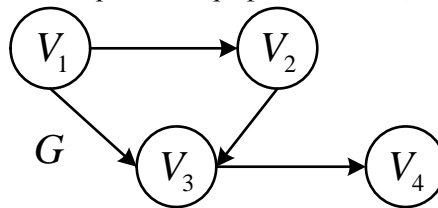


Рис. 14а. Отношение биекции для графов.

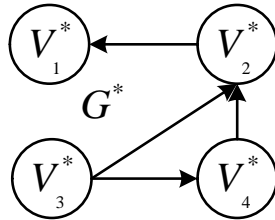


Рис. 1. 16. Отношение биекции для графов.

Биекция $\varphi: \mathbf{V} \rightarrow \mathbf{V}^*$ для данных графов выглядит следующим образом:

$$\varphi: \begin{matrix} V_1 \rightarrow V_3^* \\ V_2 \rightarrow V_4^* \\ V_3 \rightarrow V_2^* \\ V_4 \rightarrow V_1^* \end{matrix}, \text{ из этого следует, что граф } G \text{ изоморфен графу } G^* .$$

Определение. Если дуга $e = \langle V_1, V_2 \rangle$ положительно инцидентна вершине V_1 и отрицательно инцидентна вершине V_2 , то V_1 называется начальной вершиной, вершина V_2 называется конечной вершиной.

Пример 15. Рассмотрим ориентированный граф G (Рис. 15).

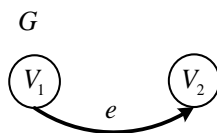


Рис. 15. Положительная и отрицательная инцидентность для графов.

Для данного графа G на Рис. 15 дуга e положительно инцидентна вершине V_1 и отрицательно инцидентна вершине V_2 , вершина V_1 является начальной вершиной, а вершина V_2 является

конечной для дуги e .

Определение. Петлей называется дуга $e = \langle V_1, V_2 \rangle$, где $V_1 = V_2 = V$. Обозначение $e = \langle V, V \rangle$. (В этом случае вершина V смежна сама с собой.)

Пример 16. Петля в графе (Рис. 16).

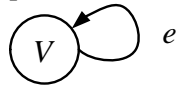


Рис. 16. Петля в графе

Рис. 1. 2. Петля в графе.

Для данного графа (Рис. 16) вершина V смежна сама с собой и дуга $e = \langle V, V \rangle$ является петлей.

Определение. Если даны дуги $e_1 = \langle V_1, V_2 \rangle$ и $e_2 = \langle V_1, V_2 \rangle$, где V_1 – начальная вершина и V_2 – конечная вершина для обеих дуг e_1 и e_2 одновременно, то дуги e_1 и e_2 называются строго параллельными.

Определение. Если даны дуги $e_1 = \langle V_1, V_2 \rangle$ и $e_2 = \langle V_2, V_1 \rangle$, то есть V_1 – начальная вершина и V_2 – конечная вершина для дуги e_1 и, наоборот, V_2 – начальная вершина и V_1 – конечная вершина для дуги e_2 , то дуги e_1 и e_2 называются нестрого параллельными.

Пример 17. Строго параллельные и нестрого параллельные дуги ориентированного графа (Рис. 17).

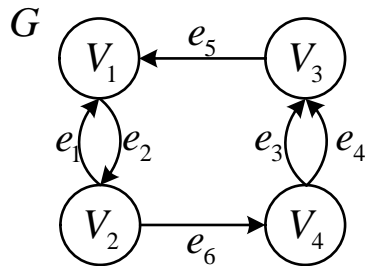


Рис. 17. Строго и нестрого параллельные дуги графа.

Дуги e_1 и e_2 нестрого параллельные, так как V_1 – начальная вершина, V_2 – конечная вершина для дуги e_2 и V_2 – начальная вершина, V_1 – конечная вершина для дуги e_1 . Докажите, что дуги e_3 и e_4 – строго параллельные.

Определение. Дуга e_1 смежна с дугой e_2 , если конечная вершина дуги e_1 совпадает с начальной вершиной дуги e_2 .

Пример 18. Смежные дуги ориентированного графа (Рис. 18).

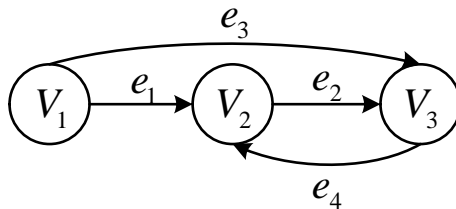


Рис. 18. Смежные дуги графа.

На Рис. 18 дуга e_1 смежна с дугой e_2 , а так же дуга e_3 смежна с дугой e_4 . Читателю предлагается найти другие пары смежных дуг?

Определение. Число дуг, положительно инцидентных вершине V , называется положительной степенью вершины V ,

обозначается $\delta^+(V)$, а число дуг, отрицательно инцидентных вершине V , называется отрицательной степенью вершины V , обозначается $\delta^-(V)$.

Определение. Степенью вершины V ориентированного графа, называется сумма положительной и отрицательной степеней вершины. Обозначение: $\delta(V) = \delta^+(V) + \delta^-(V)$.

Пример 19. Степени вершин ориентированного графа (Рис. 19).

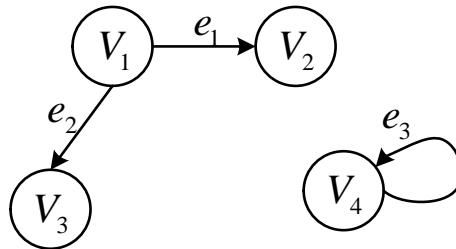


Рис. 19. Ориентированный граф.

Определим положительные, отрицательные и общие степени всех вершин данного на Рис. 19 ориентированного графа.

$$\delta^+(V_1) = 2; \delta^-(V_1) = 0; \delta(V_1) = 2;$$

$$\delta^+(V_2) = 0; \delta^-(V_2) = 1; \delta(V_2) = 1;$$

$$\delta^+(V_3) = 0; \delta^-(V_3) = 1; \delta(V_3) = 1;$$

$$\delta^+(V_4) = 1; \delta^-(V_4) = 1; \delta(V_4) = 2.$$

Утверждение: $\sum_{V \in V} \delta^+(V) = \sum_{V \in V} \delta^-(V) = |\mathbf{E}|$, где $|\mathbf{E}|$ – число дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$.

Теорема о числе вершин нечетной степени в орграфе. В орграфе число вершин нечетной степени четно.

С доказательством можно ознакомиться в предыдущей теме.

Определение. Вершина v называется изолированной, если ее степень равна 0, т.е. $\delta(v) = 0$.

Определение. Орграф называется пустым, если все его вершины являются изолированными.

Подробнее смотрите пример 4 предыдущей темы (**Ошибка! Источник ссылки не найден.**).

Определение. Граф $G_1 = \langle \mathbf{V}_1, \mathbf{E}_1 \rangle$ называется подграфом $G = \langle \mathbf{V}, \mathbf{E} \rangle$ при выполнении следующих двух условий:

- 1) $\mathbf{V}_1 \subseteq \mathbf{V}, \mathbf{E}_1 \subseteq \mathbf{E}$.
- 2) Если дуга $e \in \mathbf{E}_1$ положительно инцидентна вершине $V_1 \in \mathbf{V}_1$ и отрицательно инцидентна вершине $V_2 \in \mathbf{V}_1$, то и дуга $e \in \mathbf{E}$ также положительно инцидентна вершине $V_1 \in \mathbf{V}$ и отрицательно инцидентна вершине $V_2 \in \mathbf{V}$.

Пример 20. Пример подграфа для орграфа (Рис. 20).

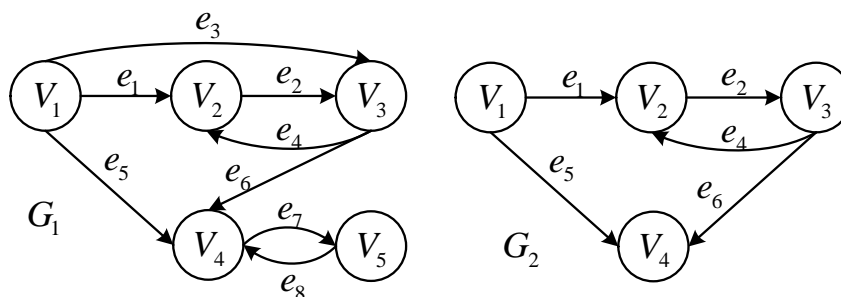


Рис. 20. Граф и подграф.

В данном случае (Рис. 20. Граф и подграф.) граф G_2 является подграфом графа G_1 . Опишем множества вершин и дуг: для орграфа $G_1 = \langle \mathbf{V}_1, \mathbf{E}_1 \rangle$, множество вершин $\mathbf{V}_1 = \{V_1, V_2, V_3, V_4, V_5\}$ и множество дуг $\mathbf{E}_1 = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$ и для орграфа $G_2 = \langle \mathbf{V}_2, \mathbf{E}_2 \rangle$ множество вершин $\mathbf{V}_2 = \{V_1, V_2, V_3, V_4\}$ и множество дуг $\mathbf{E}_2 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. Условия 1) и 2) выполняются, так как

$V_2 \subseteq V_1, E_2 \subseteq E_1$ и свойства положительной и отрицательной инцидентности выполняются.

Ориентированные маршруты, пути, контуры

Определение. Ориентированным маршрутом (или ормаршрутом) длины n называется последовательность (не обязательно различных) дуг e_1, e_2, \dots, e_n таких, что для соответствующей последовательности $n+1$ вершин V_0, V_1, \dots, V_n выполняется условие $e_i = \langle V_{i-1}, V_i \rangle, i = \overline{1, n}$. (Заметим, что нумерация говорит о последовательности дуг и вершин в ормаршруте, а не о нумерации в орграфе.)

Определение. Ормаршрут замкнут, если $V_0 = V_n$ (начальная вершина совпадает с последней вершиной). Также замкнутый ормаршрут называется циклическим ормаршрутом.

Пример 21. Пример ормаршрута и замкнутого ормаршрута для орграфа (Рис. 21).

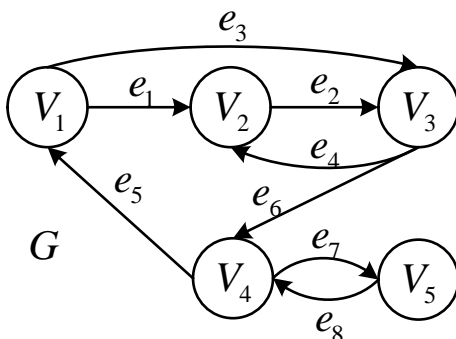


Рис. 21. Ориентированный маршрут в графе.

Рассмотрим Рис. 21, один из ормаршрутов из вершины V_1 в вершину V_5 записан в виде следующей последовательности: e_1, e_2, e_6, e_7 , последовательность вершин: V_1, V_2, V_3, V_4, V_5 . Пример

циклического ормаршрута: e_3, e_6, e_7, e_8, e_5 , этот ормаршрут идет из вершины V_1 и возвращается в вершину V_1 , последовательностью вершин: $V_1, V_3, V_4, V_5, V_4, V_1$.

Определение. Ормаршрут в котором нет повторяющихся дуг, называется путем; и простым путем, если все его вершины различны.

Определение. Замкнутый путь называется контуром. Замкнутый простой путь называется простым контуром.

Пример 22. Вернемся к примеру 21, ормаршрут e_1, e_2, e_6, e_7 является путем, но не является контуром, так как он не замкнут, а другой ормаршрут e_3, e_6, e_7, e_8, e_5 является и путем и контуром, так как замкнут, но этот ормаршрут не является простым маршрутом и простым контуром, так как проходит через вершину V_4 два раза.

Сильная связность

Определение. Орграф называется сильно связным, если для каждой пары различных вершин V_i и V_j существует путь из V_i в V_j и из V_j в V_i .

Определение. Орграф называется сильно k – связным, если для каждой пары различных вершин V_i и V_j существует по крайней мере k путей из V_i в V_j , и из V_j в V_i , которые не имеют общих вершин (а, следовательно, и дуг) за исключением V_i в V_j .

Пример 23. 3-связный орграф (Рис. 22).

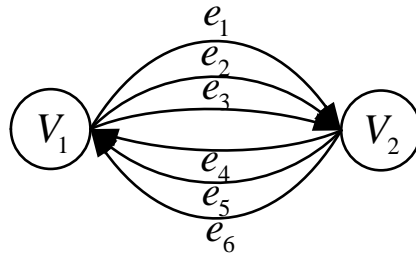


Рис. 22. Трех-связный граф.

Этот орграф (Рис. 22) является сильно связным, так как для вершин, существует путь из V_1 в V_2 и из V_2 в V_1 . Так как путей минимум по 3, этот орграф называется 3-связным.

Пример 24. Сильно связные графы.

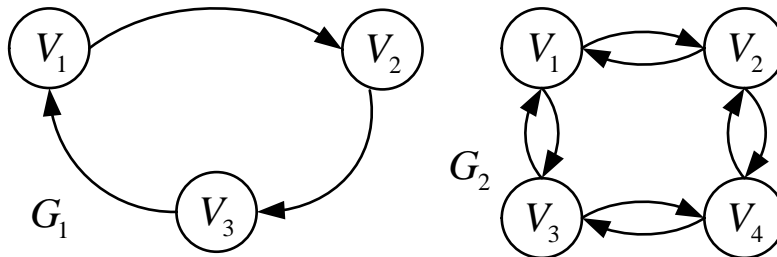


Рис. 23. Сильносвязные графы.

Графы G_1 и G_2 (Рис. 23) являются сильно связными графами. Является ли граф из примера 21 сильно связным?

Ориентированные деревья

Определение. Орграф является ориентированным деревом, растущим из корня V_0 , если:

- 1) он образует дерево в неориентированном смысле;

2) единственная цепь между V_0 и любой другой вершиной V является путем из V_0 в V .

Пример 25.

Какую дугу нужно убрать на Рис. 24, чтобы получить ориентированное дерево? (решение не единственно)

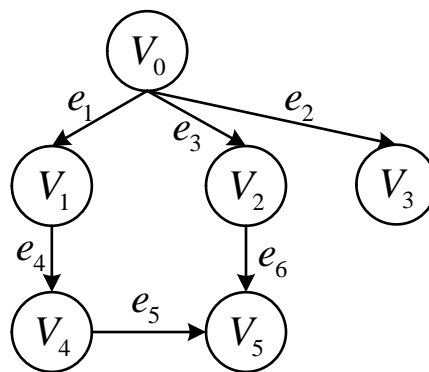


Рис. 24. Исходный граф.

Тема 3. Метрические характеристики графов. Матричное представление графов: матрица инцидентности для неорграфа, матрица смежности для неорграфа, матрица инцидентности для орграфа, матрица смежности для орграфа. Список смежности

Метрические характеристики графов

Рассмотрим связный невзвешенный неорграф $G = (\mathbf{V}, \mathbf{E})$; $V_i, V_j, V_k \in \mathbf{V}$.

Обозначения. $d(V_i, V_j)$ – длина (количество ребер) кратчайшей простой цепи между V_i и V_j , и положим, что $d(V_i, V_j) = \infty$, если V_i и V_j не соединены. Такое расстояние будет удовлетворять следующим аксиомам метрики:

- 1) $d(V_i, V_j) \geq 0$,
- 2) $d(V_i, V_j) = 0 \Leftrightarrow V_i = V_j$ ¹⁰,
- 3) $d(V_i, V_j) = d(V_j, V_i)$,
- 4) $d(V_i, V_j) + d(V_j, V_k) \geq d(V_i, V_k)$.

Определение. Эксцентриситетом фиксированной вершины V_i называется величина $l(V_i) = \max_{V_j \in \mathbf{V}} d(V_i, V_j)$.

Определение. Диаметром графа $G = (\mathbf{V}, \mathbf{E})$ называется величина $d(G) = \max_{V_i \in \mathbf{V}} l(V_i)$.

Определение. Вершина V_i называется периферийной, если $l(V_i) = d(G)$.

Определение. Радиусом графа $G = (\mathbf{V}, \mathbf{E})$ называется величина $r(G) = \min_{V_i \in \mathbf{V}} l(V_i)$.

¹⁰ Знак \Leftrightarrow означает «тогда и только тогда».

Определение. Вершина V_i называется центральной, если $l(V_i) = r(G)$.

Определение. Множество всех центральных вершин графа называется его центром.

Матрица инцидентности для неорграфа

$G = (\mathbf{V}, \mathbf{E})$ – неорграф, имеющий n вершин ($|\mathbf{V}| = n$) и m ребер ($|\mathbf{E}| = m$). $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, $\mathbf{E} = \{e_1, e_2, \dots, e_m\}$.

Матрицей инцидентности для графа $G = (\mathbf{V}, \mathbf{E})$ будет называться матрица:

$$A = [a_{i,j}]_{i=\overline{1,n}, j=\overline{1,m}} = \left[\begin{array}{c|cccc} A & e_1 & e_2 & \dots & e_m \\ \hline V_1 & & & & \\ V_2 & & a_{i,j} & & \\ \dots & & & & \\ V_n & & & & \end{array} \right],$$

где $a_{i,j} = \begin{cases} 0, & \text{если ребро } e_j \text{ не инцидентно вершине } V_i; \\ 1, & \text{если ребро } e_j \text{ инцидентно вершине } V_i; \\ 2, & \text{если ребро } e_j \text{ - петля в вершине } V_i. \end{cases}$

и $i = \overline{1,n}$, $j = \overline{1,m}$.

Свойство 1:

$$a_{\cdot j} = \sum_{i=1}^n a_{i,j} = 2,$$

$$a_{i\cdot} = \sum_{j=1}^m a_{i,j} = \delta(V_i).$$

В каждом столбце ровно по две единицы, кроме столбцов, соответствующих петлям. В столбцах, соответствующих петлям, – только одна цифра 2.

Свойство 2:

В случае, когда граф можно разбить на два или более несвязных компоненты, то матрица инцидентности будет иметь блочно-диагональную структуру при условии, что вершины первой компоненты пронумерованы первыми, второй компоненты – вторыми, и т.д. (то есть по возрастанию).

Пример 26. Матрица инцидентности для неорграфа. Рассмотрим несвязный граф Рис. 25:

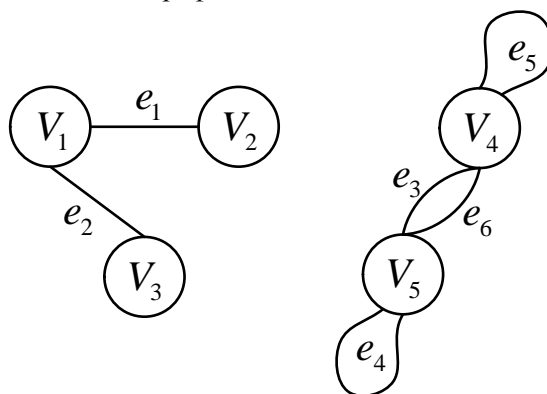


Рис. 25. Исходный неорграф.

Составим матрицу инцидентности для данного графа (Рис. 25). Так как ребро e_1 соединяет вершины V_1 и V_2 , то в первом столбце первой строке ставим единицу и в первом столбце и второй строке ставим единицу, так как ребро e_5 является петлей в вершине V_4 , то в пятом столбце и четвертой строки ставим цифру 2, и так далее:

$$A = \begin{array}{c|cccccc} A & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline V_1 & 1 & 1 & 0 & 0 & 0 & 0 \\ V_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ V_3 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline V_4 & 0 & 0 & 1 & 0 & 2 & 1 \\ V_5 & 0 & 0 & 1 & 2 & 0 & 1 \end{array}.$$

Матрица имеет блочно-диагональный вид, так как существуют две компоненты связности, а вершины и ребра пронумерованы соответствующим способом.

Матрица смежности для неорграфа

Пусть $G=(\mathbf{V},\mathbf{E})$ – неорграф, имеющий n вершин ($|\mathbf{V}|=n$, $\mathbf{V}=\{V_1,V_2,\dots,V_n\}$).

Тогда матрицей смежности для неорграфа $G=(\mathbf{V},\mathbf{E})$ будет называться матрица:

$$B=[b_{i,j}]_{i,j=\overline{1,n}} = \left[\begin{array}{c|cccc} B & V_1 & V_2 & \dots & V_n \\ \hline V_1 & & & & \\ V_2 & & b_{i,j} & & \\ \dots & & & & \\ V_n & & & & \end{array} \right],$$

где $b_{i,j} = \{ \text{число ребер одновременно инцидентных вершинам } V_i \text{ и } V_j, i, j = \overline{1,n} . \}$

Свойство. Матрица смежности для неорграфов всегда является симметричной матрицей относительно главной диагонали.

Пример 27. Составьте матрицу смежности для неорграфа. Рассмотрим следующий граф (Рис. 26):

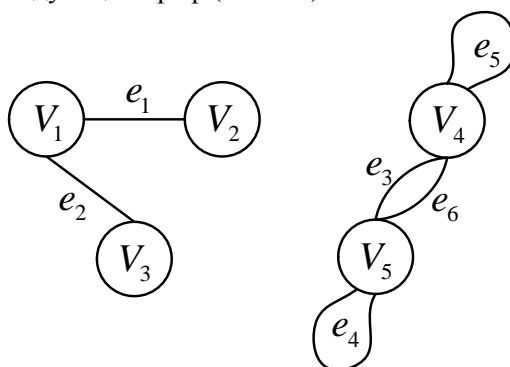


Рис. 26. Исходный неорграф.

При составлении матрицы смежности в строках и столбцах стоит нумерация по вершинам, а значениями матрицы являются величины, равные числу ребер, соединяющих соответствующие вершины. Т.е. вершины V_1 и V_2 соединены одним ребром, следовательно, в первой строке и втором столбце стоит единица, и во второй строке и первом столбце тоже единица. Отметим, что вершины V_4 и V_5 соединены двумя ребрами, следовательно, в пятой строке и четвертом столбце стоит цифра 2 и в четвертой строке и пятом столбце стоит цифра 2. Обратите внимание, хотя есть петли, в матрице смежности они обозначаются цифрой 1, т.к. одна петля.

Итак, для графа на Рис. 26, получаем матрицу смежности:

$$B = \begin{array}{c|ccccc} B & V_1 & V_2 & V_3 & V_4 & V_5 \\ \hline V_1 & 0 & 1 & 1 & 0 & 0 \\ V_2 & 1 & 0 & 0 & 0 & 0 \\ V_3 & 1 & 0 & 0 & 0 & 0 \\ \hline V_4 & 0 & 0 & 0 & 1 & 2 \\ V_5 & 0 & 0 & 0 & 2 & 1 \end{array} .$$

Матрица является симметричной относительно главной диагонали.

Матрица инцидентности для орграфа

Пусть $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – орграф, имеющий n вершин ($|\mathbf{V}| = n$) и m дуг ($|\mathbf{E}| = m$): $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, $\mathbf{E} = \{e_1, e_2, \dots, e_m\}$.

Матрицей инцидентности для орграфа $G = \langle \mathbf{V}, \mathbf{E} \rangle$ будет называться матрица:

$$A = [a_{i,j}]_{i=\overline{1,n}, j=\overline{1,m}} = \left[\begin{array}{c|cccc} A & e_1 & e_2 & \dots & e_m \\ \hline V_1 & & & & \\ V_2 & & a_{i,j} & & \\ \dots & & & & \\ V_n & & & & \end{array} \right], \text{ где}$$

$$a_{i,j} = \begin{cases} 0, & \text{если дуга } e_j \text{ не инцидентна вершине } V_i; \\ 1, & \text{если дуга } e_j \text{ положительно инцидентна} \\ & \text{вершине } V_i \text{ (т.е. выходит из вершины } V_i); \\ -1, & \text{если дуга отрицательно инцидентна} \\ & \text{вершине } V_i \text{ (т.е. входит в вершину } V_i); \\ 2, & \text{если дуга } e_j \text{ - петля в вершине } V_i. \end{cases}$$

$$i = \overline{1, n}, \quad j = \overline{1, m}.$$

Матрица смежности для орграфа

Пусть $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – орграф, имеющий n вершин ($|\mathbf{V}| = n$, $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$).

Тогда матрицей смежности для орграфа $G = (\mathbf{V}, \mathbf{E})$ будет называться матрица:

$$B = [b_{i,j}]_{i,j=\overline{1,n}} = \left[\begin{array}{c|cccc} B & V_1 & V_2 & \dots & V_n \\ \hline V_1 & & & & \\ V_2 & & b_{i,j} & & \\ \dots & & & & \\ V_n & & & & \end{array} \right],$$

где $b_{i,j}$ – число дуг, направленных от вершины V_i к вершине V_j , $i, j = \overline{1, n}$.

Пример 28. Найти матрицу инцидентности и смежности для орграфа.

Рассмотрим несвязный орграф (Рис. 27).

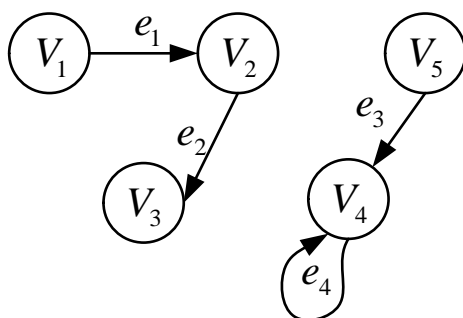


Рис. 27. Исходный оргграф.

Составим матрицу инцидентности для графа на Рис. 27.

Так как дуга e_1 направлена из вершины V_1 в V_2 , то в первой строке первого столбца стоит 1, а во второй строке первого столбца стоит -1. Петля в вершине V_4 дает в четвертой строке и четвертом столбце цифру 2, и так далее.

Матрица инцидентности для оргграфа $G = \langle \mathbf{V}, \mathbf{E} \rangle$:

$$A = \begin{bmatrix} A & e_1 & e_2 & e_3 & e_4 \\ V_1 & 1 & 0 & 0 & 0 \\ V_2 & -1 & 1 & 0 & 0 \\ V_3 & 0 & -1 & 0 & 0 \\ V_4 & 0 & 0 & -1 & 2 \\ V_5 & 0 & 0 & 1 & 0 \end{bmatrix} .$$

Аналогично строим матрицу смежности, обращая внимание, что номера строк и столбцов являются вершинами графа. Так как дуга e_1 идет из вершины V_1 в V_2 , то во втором столбце первой строки будет стоять единица, а не наоборот. На диагонали единица будет стоять только в четвертой строке и четвертом столбце, так как петля в вершине V_4 .

Итак, матрица смежности для данного оргграфа $G = \langle \mathbf{V}, \mathbf{E} \rangle$:

$$B = \begin{array}{c|ccccc} B & V_1 & V_2 & V_3 & V_4 & V_5 \\ \hline V_1 & 0 & 1 & 0 & 0 & 0 \\ V_2 & 0 & 0 & 1 & 0 & 0 \\ V_3 & 0 & 0 & 0 & 0 & 0 \\ \hline V_4 & 0 & 0 & 0 & 1 & 0 \\ V_5 & 0 & 0 & 0 & 1 & 0 \end{array}.$$

Отметим, что граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$ является несвязным, и это отражено в блочно-диагональной структуре матрицы инцидентности и смежности.

Теорема о числе ормаршрутов между двумя вершинами орграфа. Матрица смежности B^n дает число ориентированных маршрутов длины n между любыми двумя вершинами ориентированного графа. (Необязательно различные дуги.)

Доказательство. (доказательство проводится методом математической индукции)

1) Рассмотрим граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$. Пусть $|\mathbf{V}| = m$. Введем обозначения:

$b_{i,k}$ – число дуг, соединяющих вершину V_i с вершиной V_k ,

$b_{k,j}$ – число дуг, соединяющих вершину V_k с вершиной V_j ,

$b_{i,j}^{(2)}$ – число различных ориентированных маршрутов длины

2 (то есть маршрут состоит из двух дуг) от вершины V_i к вершине V_j и проходящих через вершину V_k , $k = \overline{1, m}$.

Тогда $\sum_{k=1}^m b_{i,k} \cdot b_{k,j} = b_{ij}^{(2)}$. Теорема очевидна для B^2 .

2) Пусть теорема верна для матрицы B^{n-1} . Покажем, что она верна для матрицы $B^n = B^{n-1} B$.

Если $b_{i,j}^{(n-1)}$ – число всех ормаршрутов длины $(n-1)$, соединяющих вершину V_i с вершиной V_k , а $b_{k,j}$ – число дуг,

соединяющих вершину V_k с вершиной V_j , то $b_{i,j}^{(n-1)} \cdot b_{k,j}$ – число всех ормаршрутов от вершины V_i к вершине V_j , проходящих через вершину V_k . Тогда $\sum_{k=1}^m b_{i,k}^{(n-1)} \cdot b_{k,j} = b_{i,j}^{(n)}$ – число всех ормаршрутов длины n направленных от вершины V_i к вершине V_j и $b_{i,j}^{(n)}$ – элемент матрицы B^n .

Замечание 1. Если существует $l, \forall n \geq l: B^n = 0$, то в графе нет циклов. Если в графе нет циклов, то элементы матрицы $B^k, k \in N$ дают число простых путей между любыми двумя вершинами графа.

Замечание 2. Теорема верна и для неориентированных графов.

Список смежности

Список смежности используется, например, для слабосвязных графов, для того, чтобы в матрице смежности не хранить большое число нулей.

Определение. Списком смежности вершины $V \in \mathbf{V}$ называется множество $u(V_i) = \{V_j : \langle V_i, V_j \rangle \in \mathbf{E}\}$ (для неорграфа) или $u(V_i) = \{V_j : \langle V_i, V_j \rangle \in \mathbf{E}\}$ (для орграфов).

Пример 29. Граф и его список смежности.

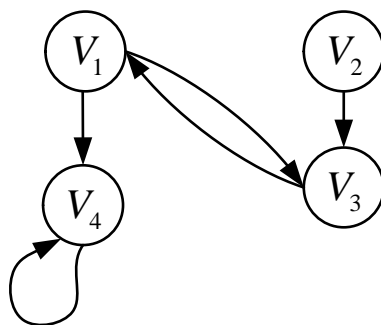


Рис. 28. Исходный граф.

Для графа, представленного на Рис. 28, получаем следующий список смежности.

$$u(V_1) = \{V_3, V_4\},$$

$$u(V_2) = \{V_3\},$$

$$u(V_3) = \{V_1\},$$

$$u(V_4) = \{V_4\}.$$

Тема 4. Построение покрывающих деревьев. Алгоритм Краскала. Построение покрывающего дерева для связного графа. Построение минимального покрывающего дерева по алгоритму Краскала. Построение максимального покрывающего дерева по алгоритму Краскала

Построение покрывающих деревьев

Напомним, что дерево T покрывает граф G , если все вершины графа G принадлежат дереву T (аналогично с лесом из нескольких деревьев). Такое дерево называется покрывающим.

На практике часто необходимо знать, связан ли данный граф. Если граф не связан, то требуется найти все его компоненты связности. Покрывающее дерево существует только для связного графа (лес для несвязного графа).

Алгоритм Краскала (1957 г.) позволяет построить такое покрывающее дерево (или лес из нескольких деревьев). Алгоритм Краскала можно представить как процесс окрашивания ребер. Ребра окрашиваются в в два цвета: синий и оранжевый. или являются петлями Рис. 29.

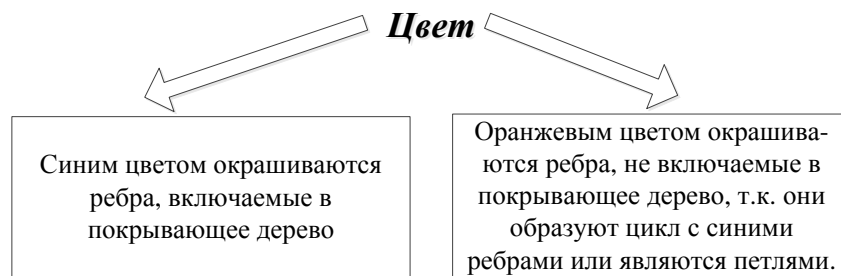


Рис. 29. Раскраска графа по алгоритму Краскала.

Сделаем некоторые подготовительные шаги для алгоритма Краскала.

- 1) а) Пронумеруем все ребра графа следующим образом:
 $e_1, e_2, \dots, e_m; |\mathbf{E}| = m$,
 б) или упорядочим ребра в лексикографическом порядке
 через запись их по вершинам:
 $(V_1, V_2), \dots, (V_1, V_5), \dots, (V_2, V_5), \dots$, то есть (V_i, V_j) , где $i \leq j$,
 $i, j = \overline{1, |\mathbf{V}|}$. Например, сначала идет ребро (V_i, V_5) , потом
 ребро (V_i, V_6) .
- 2) Образует упорядоченное множество
 $\mathbf{E} = \{(V_i, V_j) \mid i \leq j, i, j = \overline{1, |\mathbf{V}|}\}$. Лексикографический
 порядок, нумерация по вершинам.

Определение. Букет – множество вершин, принадлежащих одной компоненте связности.

Замечание. Ребро образует цикл с ребрами, уже включенными в дерево, если обе его концевые вершины принадлежат одной и той же компоненте связности.

Существует 2 случая связности графа:

- 1° Если $G = (\mathbf{V}, \mathbf{E})$ – связный граф, где $|\mathbf{V}| = n$, то построение покрывающего дерева по алгоритму Краскала заканчивается, когда количество ребер, окрашенных в синий цвет, становится равным $(n - 1)$.
- 2° Если $G = (\mathbf{V}, \mathbf{E})$ – несвязный граф, где $|\mathbf{V}| = n$, то построение покрывающего дерева по алгоритму Краскала заканчивается после раскраски всех ребер графа. Число покрывающих деревьев в таком лесу будет равно числу букетов.

Приведем пример практической задачи, сводящейся к вопросу о связности графа.

Задача. В небольшой деревушке некоторые из жителей имеют каждодневные встречи друг с другом. Может ли в этой деревне распространиться какой-либо слух?

Чтобы ответить на этот вопрос, поставим в соответствие каждому жителю деревни вершину графа. Соединим две вершины

ребром, если соответствующие жители ежедневно общаются друг с другом. При условии связности полученного таким образом графа на поставленный в задаче вопрос можно ответить положительно.

Алгоритм Краскала. Построение покрывающего дерева для связного графа

Ниже пошагово представлен алгоритм построения покрывающего дерева, предложенный Джозефом Краскалом в 1957 г.

Начало: Все рёбра графа $G = (V, E)$ не окрашены и ни один из букетов не сформирован.

Шаг 1. Все петли окрасить в оранжевый цвет.

Шаг 2. Из упорядоченного множества E выбирается первое ребро, не являющееся петлей. Это ребро окрашивается в синий цвет и формируется букет, в который включаются концевые вершины выбранного ребра.

Шаг 3. Из оставшихся ребер выбирается первое неокрашенное ребро, которое не является петлей. Если в графе такого ребра нет, следует закончить процедуру и перейти к шагу 3.

После выбора ребра возможны 4 случая:

- А. Обе концевые вершины выбранного ребра принадлежат одному и тому же букету. В этом случае ребро окрашивается в оранжевый цвет.
- Б. Одна из концевых вершин ребра принадлежит существующему букету, а другая – не принадлежит ни одному из уже сформированных букетов. В этом случае ребро окрашивается в синий цвет, и его вторая концевая вершина включается в букет, которому принадлежит первая концевая вершина.
- В. Концевые вершины выбранного ребра принадлежат различным букетам. В этом случае ребро окрашивается в синий цвет, а оба букета, которым принадлежат его концевые вершины, объединяем в

новый букет с меньшей нумерацией.

Г. Ни одна из концевых вершин не принадлежит ни одному из сформированных букетов. В этом случае ребро окрашивается в синий цвет и формируется новый букет из концевых вершин этого ребра.

Шаг 4. Если все ребра окрашены, следует закончить процедуру. Синие ребра образуют покрывающее дерево (лес). В противном случае вернуться к началу шага 2.

Конец алгоритма.

Для понимания этого алгоритма приведем пример.

Пример 30. Построить покрывающее дерево по алгоритму Краскала по нумерации для графа на Рис. 30.

Рассмотрим следующий граф $G = (V, E)$ Рис. 30:

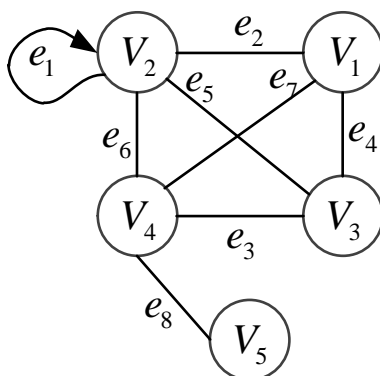


Рис. 30. Исходный граф.

Упорядочим множество ребер: $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$.

Для поиска покрывающего дерева по алгоритму Краскала необходимо пошагово расписать процесс окраски графа. Представим процесс окрашивания в виде таблицы. Ребра окрашиваем по нумерации Табл. 1.

Табл. 1. Процесс окраски графов пошагово.

Ребро	Цвет	Букет №1	Букет №2
\emptyset	–	Пуст	Пуст
e_1	Оранжевый	Пуст	Пуст
e_2	Синий (1)	$\{V_1, V_2\}$	Пуст
e_3	Синий (2)	$\{V_1, V_2\}$	$\{V_3, V_4\}$
e_4	Синий (3)	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_5	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_6	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_7	Оранжевый	$\{V_1, V_2, V_3, V_4\}$	Пуст
e_8	Синий (4)	$\{V_1, V_2, V_3, V_4, V_5\}$	Пуст

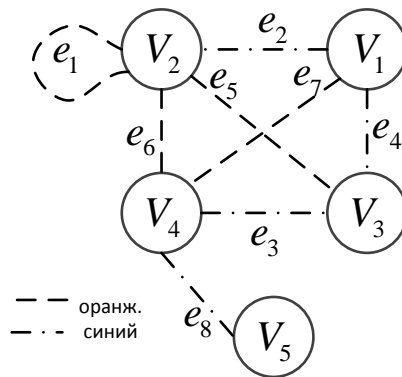


Рис. 31. Результат раскраски графа по алгоритму Краскала.

Ответ: $T = \{e_2, e_3, e_4, e_8\}$. Эти ребра образуют покрывающее дерево Рис. 32.

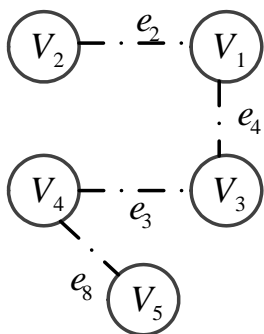


Рис. 32. Покрывающее дерево, построенное по алгоритму Краскала.

Замечание. Представленный алгоритм Краскала не учитывает веса ребер и используется только для невзвешенных графов.

Определение. Псевдографом называется граф, в котором допускаются петли и кратные параллельные ребра.

Пример 31. Пример псевдографа Рис. 33.

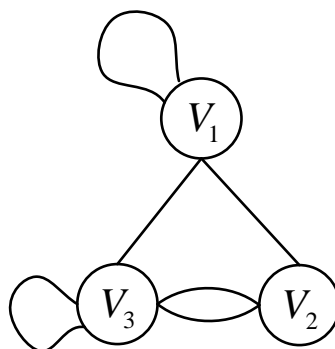


Рис. 33. Псевдограф.

Определение. Мультиграфом называется псевдограф без петель.

Пример 32. Пример мультиграфа Рис. 34.

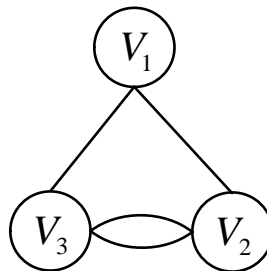


Рис. 34. Мультиграф.

Определение. Неорграф называется простым, если он не содержит петель и кратных параллельных ребер.

Пример 33. Пример простого неорграфа Рис. 35.

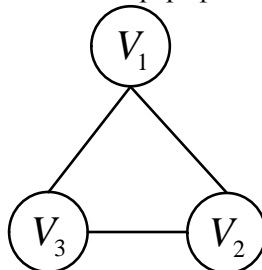


Рис. 35. Простой неорграф.

Определение. Граф называется планарным (плоским), если он может быть изображен на плоскости так, что все пересечения ребер являются его вершинами.

Пример 34. Примеры планарных и не планарных графов.

- а) На Рис. 36 граф G_1 является планарным, так как для графа G_1 существует изоморфный ему граф G_2 , который является планарным.

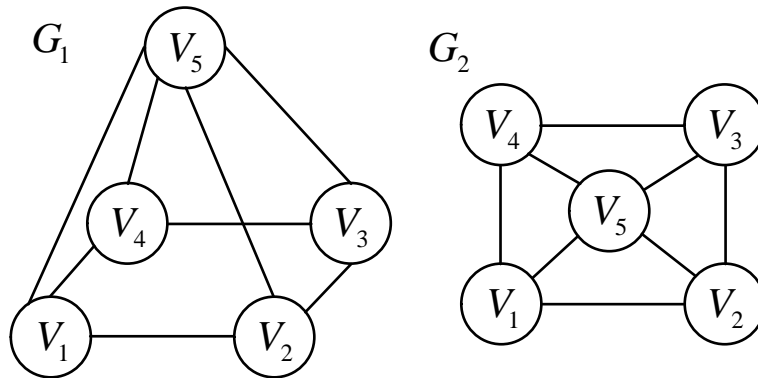


Рис. 36. Планарные графы.

- б) На Рис. 37 при добавлении к графу G_1 дополнительной дуги, получим граф G_3 , который также является планарным, и представляется на плоскости изоморфным графом G_4 .

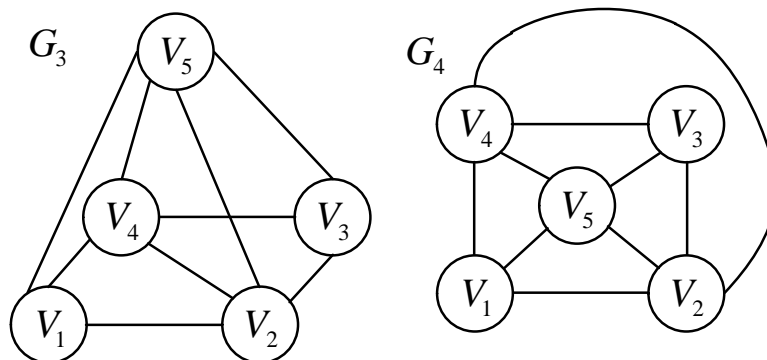


Рис. 37. Планарные графы.

- с) Граф G_5 Рис. 38 не является планарным, так как его нельзя представить на плоскости в виде графа без пересечений ребер.

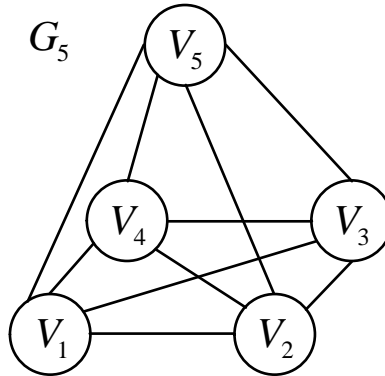


Рис. 38. Не планарный граф.

- d) На Рис. 39 граф G_6 является планарным (звезда), так как его можно представить в виде изоморфного графа G_7 .

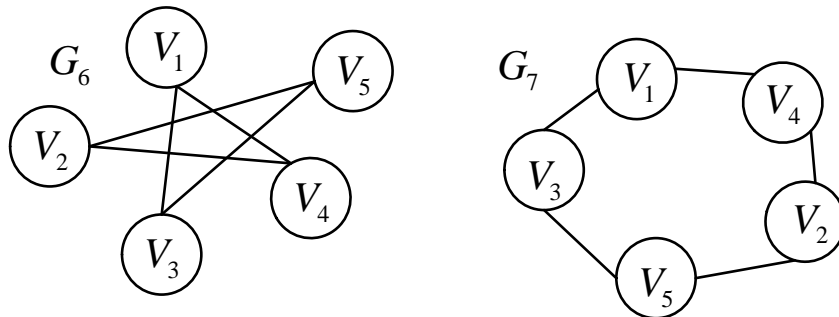


Рис. 39. Изоморфные планарные графы.

Определение. Для графа $G=(\mathbf{V},\mathbf{E})$, где $|\mathbf{V}|=n$, матрица весов \mathbf{W} определяется следующим образом:

$$\mathbf{W} = \left(\begin{array}{c|ccc} & V_1 & \dots & V_n \\ \hline V_1 & & & \\ \dots & & & \\ V_n & & & \end{array} \right), \quad i, j = \overline{1, n} \text{ и}$$

$$w_{i,j} = \begin{cases} \text{минимальный вес ребра от вершины } V_i \text{ до вершины } V_j, \\ 0, \text{ если } i = j, \\ \infty, \text{ если ребра } (V_i, V_j) \text{ не существует.} \end{cases}$$

Построение минимального покрывающего дерева по алгоритму Краскала

Данный алгоритм применяется для взвешенных графов. Ребра упорядочиваются по возрастанию весов и по нумерации для ребер с одинаковыми весами. Все правила, введенные в предыдущем разделе для данного алгоритма верны. Для каждого минимального дерева в качестве ответа может быть подсчитан вес дерева $\mathbf{W}_{T_{\min}}$, который складывается из весов всех входящих в дерево ребер.

Пример 35. Построить минимальное покрывающее дерево по алгоритму Краскала для графа на Рис. 40. Найти вес дерева T ($\mathbf{W}_{T_{\min}}$).

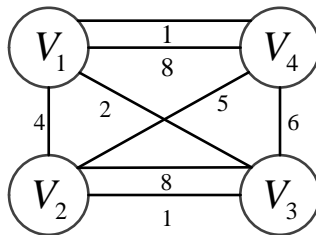


Рис. 40. Исходный граф.

Упорядочим множество ребер \mathbf{E} по возрастанию весов:

$$\mathbf{E} = \{(V_2, V_3), (V_1, V_3), (V_1, V_2), (V_2, V_4), (V_3, V_4), (V_1, V_4)\}.$$

Для поиска покрывающего дерева по алгоритму Краскала, необходимо пошагово окрасить ребра графа. Представим процесс окрашивания в виде Табл. 2. Проход алгоритма Краскала. Ребра окрашиваем по нумерации.

Табл. 2. Проход алгоритма Краскала.

Ребро	Цвет	Букет №1
\emptyset	–	Пуст
$\overset{1}{(V_2, V_3)}$	Синий (1)	$\{V_2, V_3\}$
$\overset{2}{(V_1, V_3)}$	Синий (2)	$\{V_1, V_2, V_3\}$
$\overset{4}{(V_1, V_2)}$	Оранжевый	$\{V_1, V_2, V_3\}$
$\overset{5}{(V_2, V_4)}$	Синий (3)	$\{V_1, V_2, V_3, V_4\}$

На Рис. 41 представлена раскраска графа.

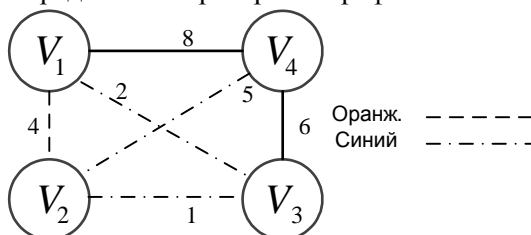


Рис. 41. Раскраска графа по алгоритму Краскала.

Ответ: Получаем минимальное покрывающее дерево $T = \{(V_2, V_3), (V_1, V_3), (V_2, V_4)\}$ Рис. 42 и вес дерева $W_{T_{\min}} = 1 + 2 + 5 = 8$

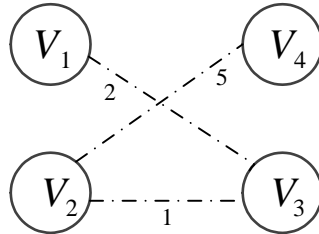


Рис. 42. Покрывающее дерево для графа.

Построение максимального покрывающего дерева по алгоритму Краскала

Следует отметить, что алгоритм применяется для взвешенных графов. Ребра упорядочиваются по убыванию весов, а затем и по нумерации для ребер с одинаковым весом. Для каждого максимального дерева в качестве ответа может быть посчитан вес дерева $W_{T_{\max}}$, который складывается из весов всех входящих в дерево ребер.

Пример 36. Построить максимальное покрывающее дерево по алгоритму Краскала для графа на Рис. 43. Найти вес дерева T ($W_{T_{\max}}$).

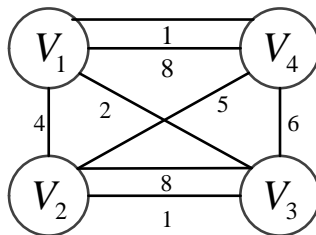


Рис. 43. Исходный граф.

Упорядочим множество ребер E по убыванию весов:
 $E = \{(V_1, V_4), (V_2, V_3), (V_3, V_4), (V_2, V_4), (V_1, V_2), (V_1, V_3), (V_1, V_4), (V_2, V_3)\}$.

Для поиска покрывающего дерева по алгоритму Краскала

необходимо пошагово расписать процесс окраски графа.
Представим процесс окрашивания в виде Табл. 3.

Табл. 3. Поиск покрывающего дерева по алгоритму Краскала.

Ребро	Цвет	Букет №1
\emptyset	–	Пуст
$\overset{8}{(V_1, V_4)}$	Синий (1)	$\{V_1, V_4\}$
$\overset{6}{(V_3, V_4)}$	Синий (2)	$\{V_1, V_3, V_4\}$
$\overset{5}{(V_2, V_4)}$	Синий (3)	$\{V_1, V_2, V_3, V_4\}$

После окраски трех ребер алгоритм окончен, т.к. букет содержит все четыре вершины. На Рис. 44Рис. 51 представлена раскраска графа.

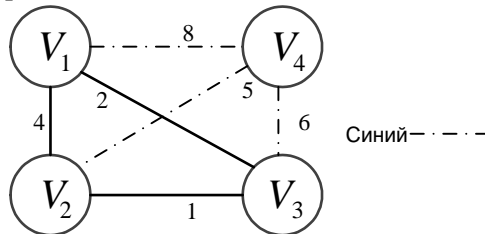


Рис. 44. Раскраска графа.

Ответ: Получаем максимальное покрывающее дерево $T = \{(V_1, V_4), (V_2, V_4), (V_3, V_4)\}$ и вес дерева $\mathbf{W}_{T_{\max}} = 8 + 5 + 6 = 19$.

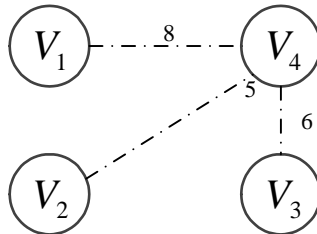


Рис. 45. Построенное по алгоритму Краскала покрывающее дерево.

Тема 5. Построение минимального покрывающего дерева для связного взвешенного графа по алгоритму Прима. Построение максимального покрывающего дерева по алгоритму Прима

Построение минимального покрывающего дерева для связного взвешенного графа по алгоритму Прима

Алгоритм Прима — алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа. Алгоритм впервые был открыт в 1930 году чешским математиком Войцехом Ярником, позже переоткрыт Робертом Примом в 1957 году, и, независимо от них, Э. Дейкстрой в 1959 году. Алгоритм очень похож на алгоритм Дейкстры. Так же этот алгоритм называется алгоритмом поиска соседа.

Будем считать, что в данном алгоритме букет будет только один, этот букет будет включать в себя вершины дерева. Вводится дополнительное множество \mathbf{E}' (на начальном этапе совпадающее с отсортированным множеством ребер \mathbf{E}), из которого мы будем удалять рассмотренные ребра.

Начало: Дан граф $G = (\mathbf{V}, \mathbf{E})$ — связный и взвешенный. Минимальное покрывающее дерево T не построено. Все ребра упорядочены по возрастанию весов и по нумерации для ребер с одинаковыми весами. Букет не сформирован. Петли удалены. Из кратных ребер оставляем ребро с минимальным весом. При удалении из множества \mathbf{E} петель и кратных ребер получаем множество \mathbf{E}' .

Шаг 1. Выбираем из упорядоченного множества \mathbf{E}' первое ребро (V_i, V_j) и включаем его в дерево T . Обе концевые вершины включаются в букет $\{V_i, V_j\}$.
 $\mathbf{E}' := \mathbf{E}' \setminus \{(V_i, V_j)\}$.

Шаг 2. Из множества \mathbf{E}' выбираем первое ребро (V_i, V_j) , инцидентное какой-либо вершине из сформированного букета.

После выбора ребра возможны 2 случая:

А. Ребро (V_i, V_j) составляет цикл с существующими ребрами дерева. Тогда $\mathbf{E}' := \mathbf{E}' \setminus \{V_i, V_j\}$ и возвращаемся к началу шага 2.

Б. Ребро (V_i, V_j) не образует с существующими ребрами дерева цикл. Тогда новая вершина добавляется к вершинам сформированного букета. $\mathbf{E}' := \mathbf{E}' \setminus \{V_i, V_j\}$. Ребро добавляется в дерево. Далее возвращаемся к началу шага 2.

Шаг 3. Если дерево (букет) включает в себя все вершины графа, то минимальное покрывающее дерево построено. (Если дерево не включает в себя все вершины, то граф не является связным.)

Конец алгоритма.

Пример 37. Построить минимальное покрывающее дерево по алгоритму Прима для графа представленного на Рис. 46. Исходный граф. Найти вес дерева T ($\mathbf{W}_{T_{\min}}$).

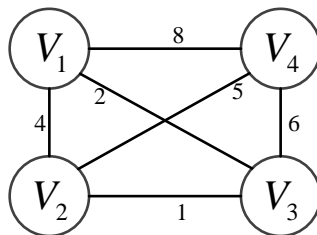


Рис. 46. Исходный граф.

Упорядочим множество ребер \mathbf{E} по возрастанию весов:

$$\mathbf{E} = \{(V_2, V_3)^1, (V_1, V_3)^2, (V_1, V_2)^4, (V_2, V_4)^5, (V_3, V_4)^6, (V_1, V_4)^8\}.$$

Для поиска минимального покрывающего дерева по алгоритму Прима необходимо пошагово расписать процесс формирования букета и изменений множества \mathbf{E}' . Представим в виде Табл. 4 пошаговое исполнение алгоритма.

Табл. 4. . Пошаговое выполнение алгоритма Прима.

Ребро	Букет	\mathbf{E}'
$(V_2, V_3)^1$	$\{V_2, V_3\}$	$\{(V_1, V_3)^2, (V_1, V_2)^4, (V_2, V_4)^5, (V_3, V_4)^6, (V_1, V_4)^8\}$
$(V_1, V_3)^2$	$\{V_1, V_2, V_3\}$	$\{(V_1, V_2)^4, (V_2, V_4)^5, (V_3, V_4)^6, (V_1, V_4)^8\}$
$(V_1, V_2)^4$	$\{V_1, V_2, V_3\}$	$\{(V_2, V_4)^5, (V_3, V_4)^6, (V_1, V_4)^8\}$
цикл		
$(V_2, V_4)^5$	$\{V_1, V_2, V_3, V_4\}$	$\{(V_3, V_4)^6, (V_1, V_4)^8\}$

Конец алгоритма.

На Рис. 47 представлены ребра графа, входящие в дерево.

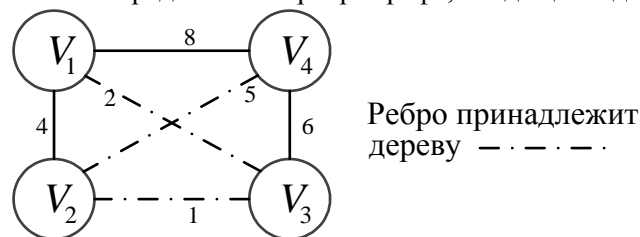


Рис. 47. Для исходного графа отмечены ребра, принадлежащие дереву.

Ответ: Получаем минимальное покрывающее дерево
 $T = \{(V_2, V_3), (V_1, V_3), (V_2, V_4)\}$ (Рис. 48) и вес дерева
 $W_{T_{\min}} = 1 + 2 + 5 = 8$.

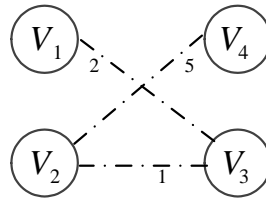


Рис. 48. Покрывающее дерево.

Построение максимального покрывающего дерева по алгоритму

Прима

Построение максимального покрывающего дерева по алгоритму Прима отличается от построения минимального покрывающего дерева только тем, что :

- 1) Ребра упорядочиваются по убыванию весов и по нумерации для ребер с одинаковым весом.
- 2) Множество E' получается из множества E путем исключения петель и кратных ребер (из кратных ребер оставляем ребро с максимальным весом).

Пример 38.

Построить максимальное покрывающее дерево по алгоритму Прима для графа на Рис. 49. Найти вес дерева T ($W_{T_{\max}}$).

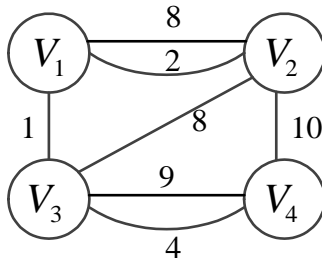


Рис. 49. Исходный граф.

Упорядочим множество ребер E по убыванию весов:
 $E = \{(V_2, V_4)^{10}, (V_3, V_4)^9, (V_1, V_2)^8, (V_2, V_3)^8, (V_3, V_4)^4, (V_1, V_2)^2, (V_1, V_3)^1\}$.

Для поиска максимального покрывающего дерева по алгоритму Прима необходимо пошагово описать процесс формирования букета и изменений множества E' . Представим в виде Табл. 5. Пошаговое выполнения алгоритма Прима.пошаговое исполнение алгоритма.

Табл. 5. Пошаговое выполнения алгоритма Прима.

Ребро	Букет	E'
$(V_2, V_4)^{10}$	$\{V_2, V_4\}$	$\{(V_3, V_4)^9, (V_1, V_2)^8, (V_2, V_3)^8, (V_3, V_4)^4, (V_1, V_2)^2, (V_1, V_3)^1\}$
$(V_3, V_4)^9$	$\{V_2, V_3, V_4\}$	$\{(V_1, V_2)^8, (V_2, V_3)^8, (V_3, V_4)^4, (V_1, V_2)^2, (V_1, V_3)^1\}$
$(V_1, V_2)^8$	$\{V_1, V_2, V_3, V_4\}$	$\{(V_2, V_3)^8, (V_3, V_4)^4, (V_1, V_2)^2, (V_1, V_3)^1\}$

Конец алгоритма.

На Рис. 50 представлены ребра графа, входящие в дерево.

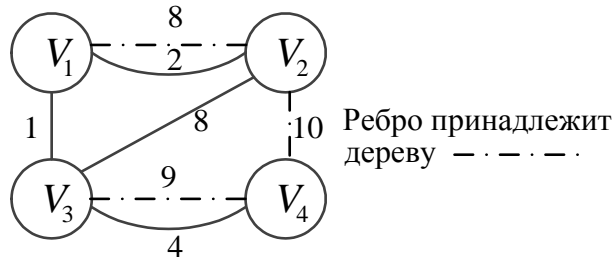


Рис. 50. Ребра, принадлежащие покрывающему дереву.

Ответ: Получаем максимальное покрывающее дерево $T = \{(V_2, V_4)^{10}, (V_3, V_4)^9, (V_1, V_2)^8\}$ (Рис. 51) и вес дерева $\mathbf{W}_{T_{\max}} = 10 + 9 + 8 = 27$.

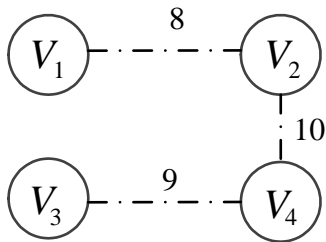


Рис. 51. Покрывающее дерево.

Тема 6. Поиск пути наименьшей длины в графе. Алгоритм Дейкстры

Поиск пути наименьшей длины в графе

Этот алгоритм первым предложил нидерландский ученый Эдсгер Дейкстра в 1959 г. Он применяется при решении задачи о кратчайших путях для ориентированного взвешенного графа.

Пусть $G = \langle V, E \rangle$ – взвешенный орграф без петель. Алгоритм дает возможность определить длину ориентированного маршрута. В случае неорграфа каждое ребро можно заменить на две дуги в одном и обратном направлении с одинаковым весом и применить алгоритм.

Применение алгоритма можно встретить в следующих задачах:

- 1) Найти путь наименьшей стоимости от пункта А до пункта В.
- 2) Пользуясь метро, доехать от пункта А до пункта В, и найти оптимальный (то есть минимальный) по времени маршрут.

Для поиска пути наименьшей длины между вершинами V_S (начало) и V_T (конец), от вершины V_S ищем расстояние до всех вершин графа $l(V_S, V)$, где $l(V_S, V_S) = 0$, и $l(V_S, V) = \infty$ тогда и только тогда, когда вершины V_S и V не соединены дугой (можно использовать матрицу весов). Каждая вершина на каждом шаге получает метку следующим образом (Рис. 52):

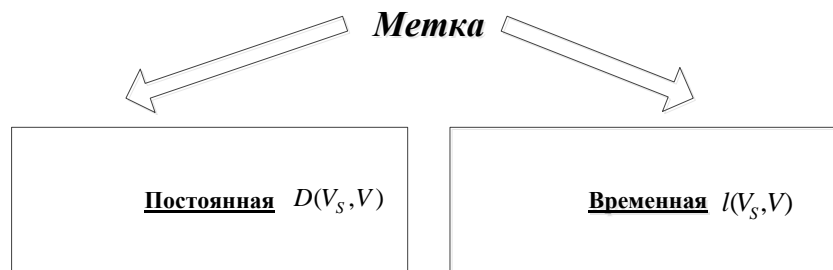


Рис. 52. Расстановка меток.

$D(V_S, V)$ – длина кратчайшего пути от вершины V_S до вершины V (постоянная метка),

$l(V_S, V)$ – длина кратчайшего пути от вершины V_S до вершины V , проходящего через вершины с постоянными метками (временная метка).

Алгоритм Дейкстры

Начало: Пусть V_S – начальная вершина. Присваиваем ей постоянную метку. $V_u := V_S$. $D(V_S, V_u) := 0$, V_u – текущая вершина.

Шаг 1. Все вершины графа кроме начальной вершины включаем в множество $\mathbf{P} := \mathbf{V} \setminus \{V_u\}$ и расставляем временные метки:

$l(V_S, V_i) := \min(w_{V_S, V_i})$, $V_i \in \mathbf{P}$. Сравниваем все значения временных меток $l(V_S, V_i)$ и выделяем $l_{\min} = l(V_S, V_j) = \mathbf{X}$.

Если две вершины имеют одинаковое минимальное значение, то для присвоения постоянной метки выбираем вершину с меньшей нумерацией. Вершине V_j присваивается постоянная метка. Вершина V_j становится текущей вершиной. $V_u := V_j$; $D(V_S, V_u) := \mathbf{X}$.

- 1) Если $V_u = V_T$, то алгоритм закончен. $D(V_S, V_T) = \mathbf{X}$.
- 2) Если $V_u \neq V_T$, то переходим к шагу 2.

Шаг 2. $\mathbf{P} := \mathbf{P} \setminus \{V_u\}$. Расставляем временные метки:

$$l(V_S, V_i) = \min(l(V_S, V_i); D(V_S, V_u) + \min(w_{V_u, V_i})),$$

$\forall V_i \in \mathbf{P}$. $l(V_S, V_i)$ – старый результат из предыдущего шага, $D(V_S, V_u)$ – длина через новую постоянную метку, w_{V_u, V_i} – вес ребра.

Сравниваем значения $l(V_S, V_i)$ и выделяем $l_{\min} = l(V_S, V_j) = \mathbf{Y}$. Если две вершины имеют одинаковый минимальный вес, то для присвоения постоянной метки выбираем вершину с меньшей нумерацией. Вершине V_j присваивается постоянная метка $V_u := V_j$; $D(V_S, V_u) = \mathbf{Y}$.

- 1) Если $V_u = V_T$, то алгоритм закончен. $D(V_S, V_T) = \mathbf{Y}$. Переходим к шагу 3.
- 2) Если $V_u \neq V_T$, то следует вернуться к началу шага 2.

Шаг 3. Длина $D(V_S, V_T)$ найдена. С помощью постоянных меток определяем путь от V_S к V_T .

Конец алгоритма.

Замечание. Алгоритм может быть использован для нахождения кратчайшего пути в графе, ребрам (дугам) которого не присвоены веса. Необходимо каждому ребру (дуге) присвоить вес, равный единице.

Пример 39. По алгоритму Дейкстры найти минимальное расстояние от вершины V_1 до вершины V_3 . Определить путь от вершины V_1 до вершины V_3 . Граф изображен на Рис. 53.

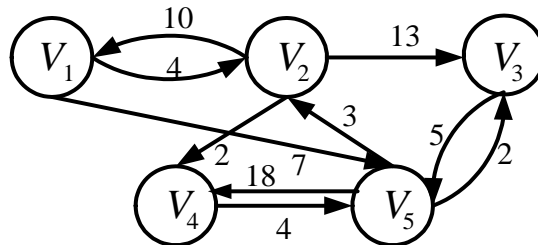


Рис. 53. Исходный граф для примера.

Начало: $V_S = V_1$ – начальная вершина и $V_T = V_3$ – конечная вершина. $\mathbf{V} = \{V_1, V_2, V_3, V_4, V_5\}$ Присваиваем постоянную метку. $V_u := V_1$. $D(V_1, V_u) := 0$.

Шаг 1. $\mathbf{P} = \mathbf{V} \setminus \{V_u\} = \mathbf{V} \setminus \{V_1\} = \{V_2, V_3, V_4, V_5\}$, Находим все $l(V_1, V_i)$:

$$l(V_1, V_2) = 4,$$

$$l(V_1, V_3) = \infty,$$

$$l(V_1, V_4) = \infty,$$

$$l(V_1, V_5) = 7.$$

Сравниваем все временные метки $l(V_1, V_i)$ и выделяем $l_{\min} = l(V_1, V_2) = 4$. Следовательно, вершине V_2 присваивается постоянная метка $V_u := V_2$;
 $D(V_1, V_2) = D(V_1, V_u) = 4$.

Шаг 2. $\mathbf{P} = \mathbf{P} \setminus \{V_u\} = \{V_3, V_4, V_5\}$. Расставляем временные метки:

$$l(V_1, V_i) = \min(l(V_1, V_i), D(V_1, V_u) + w_{V_u V_i}) \text{ для любого } V_i \in \mathbf{P}.$$

Где $l(V_1, V_i)$ – старый результат из предыдущего шага, $D(V_1, V_u)$ – длина через новую постоянную метку, $w_{V_u V_i}$ – вес.

$$\begin{aligned} l(V_1, V_3) &= \min\{l(V_1, V_3); D(V_1, V_2) + \min(w_{V_2, V_3})\} = \\ &= \min\{\infty; 4 + 13\} = 17 \end{aligned}$$

$$\begin{aligned}
l(V_1, V_4) &= \min \left\{ l(V_1, V_4); D(V_1, V_2) + \min(w_{V_2, V_4}) \right\} = \\
&= \min \{ \infty; 4 + 2 \} = 6 \\
l(V_1, V_5) &= \min \left\{ l(V_1, V_5); D(V_1, V_2) + \min(w_{V_2, V_5}) \right\} = \\
&= \min \{ 7; 4 + \infty \} = 7 \\
l_{\min} &= l(V_1, V_4) = 6 \text{ и } V_u := V_4; D(V_1, V_4) = D(V_1, V_u) = 6.
\end{aligned}$$

Шаг 3. $\mathbf{P} = \mathbf{P} \setminus \{V_u\} = \{V_3, V_5\}$.

$$\begin{aligned}
l(V_1, V_3) &= \min \left\{ l(V_1, V_3); D(V_1, V_4) + \min(w_{V_4, V_3}) \right\} = \\
&= \min \{ 17; 6 + \infty \} = 17 \\
l(V_1, V_5) &= \min \left\{ l(V_1, V_5); D(V_1, V_4) + \min(w_{V_4, V_5}) \right\} = \\
&= \min \{ 7; 6 + 4 \} = 7 \\
l_{\min} &= l(V_1, V_5) = 7 \text{ и } V_u := V_5; D(V_1, V_5) = D(V_1, V_u) = 7.
\end{aligned}$$

Шаг 4. $\mathbf{P} = \mathbf{P} \setminus \{V_u\} = \{V_3\}$.

$$\begin{aligned}
l(V_1, V_3) &= \min \left\{ l(V_1, V_3); D(V_1, V_5) + \min(w_{V_5, V_3}) \right\} = \\
&= \min \{ 17; 7 + 2 \} = 9 \\
l_{\min} &= l(V_1, V_3) = 9 \text{ и } V_u := V_3; D(V_1, V_3) = 9. \text{ Алгоритм} \\
&\text{окончен.}
\end{aligned}$$

Ищем путь по постоянным меткам: $V_1 \rightarrow V_5 \rightarrow V_3$ (Табл. 6).

Табл. 6. Поиск пути по постоянным меткам.

V_1	V_2	V_3	V_4	V_5
0	∞	∞	∞	∞
	4	∞	∞	7
		17	6	7
		17		7
		9		

Ответ: путь $V_1 \rightarrow V_5 \rightarrow V_3$ и расстояние 9.

Тема 7. Эйлеровы графы. Алгоритм поиска эйлерова цикла в графе

Эйлеровы графы

*Начало теории графов как раздела математики связывают с так называемой задачей о кенигсбергских мостах. Задача о кёнигсбергских мостах. Город Кенигсберг (**Калининград, Россия**) был построен на берегах и двух островах реки Преголи (нем. Pregel — Прегель). В городе было семь мостов, которые соединяли острова между собой и с береговыми частями города (Рис. 54). Вопрос читателю: **можно ли, выйдя из дома, вернуться обратно, пройдя по каждому мосту только один раз?***

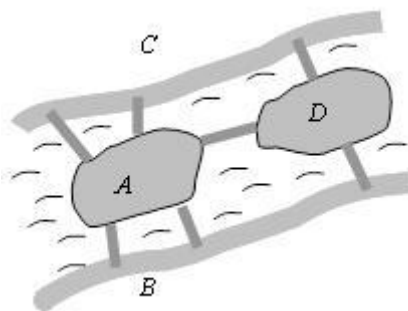


Рис. 54. Калининградские мосты.

На языке графов: Сопоставим рисунку граф (Рис. 54), вершины которого соответствуют четырем разделяемым рекой участкам суши A,B,C,D, а ребра – мостам (Рис. 55).

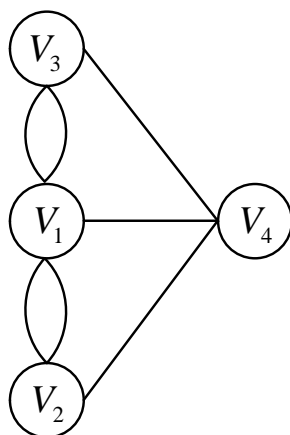


Рис. 55. Граф схемы мостов г. Калининграда.

Вопрос задачи: Существует ли цикл, содержащий все ребра графа? Возможно ли, выйдя из одной вершины, пройдя однократно по всем ребрам, вернуться обратно в ту же самую вершину?

Именно Эйлер доказал неразрешимость этой задачи.

Определение. Цикл в графе называется эйлеровым, если он содержит все ребра графа.

Определение. Связный граф, в котором есть эйлеров цикл называется эйлеровым графом.

Пример 40. Примеры эйлеровых планарных графов Рис. 56.

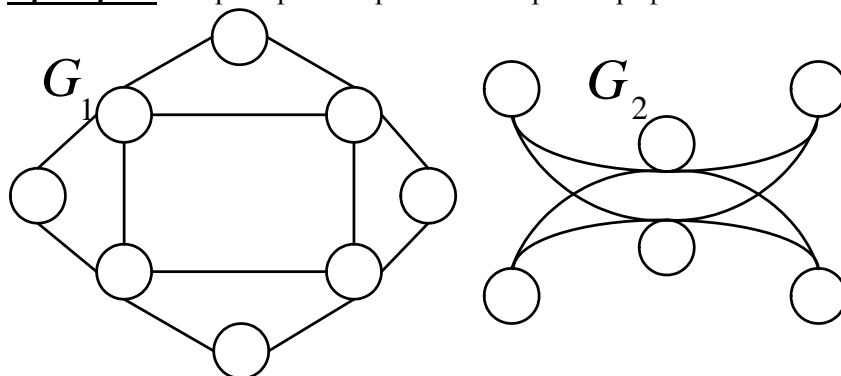


Рис. 56. Примеры эйлеровых планарных графов.

Граф G_2 можно представить в виде планарного графа (Рис. 56).

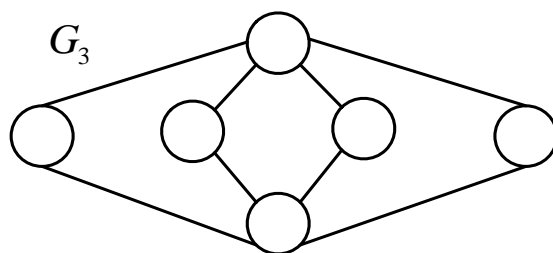


Рис. 57. Граф, изоморфный графу на рис. 56.

Теорема о четности степеней вершин в эйлеровом графе.

Связный граф является эйлеровым тогда и только тогда, когда степени всех его вершин четны.

Доказательство.

- 1) Необходимость (слева направо). Пусть граф G – эйлеров. Докажем, что граф связный и степени всех его вершин четны. Эйлеров цикл этого графа а) содержит все ребра графа (следовательно, граф связный) и б) входит в вершину по одному ребру, а выходит по другому ребру. Каждая вершина такого графа инцидентна четному количеству ребер, следовательно, степень каждой вершины четна (т.к. ребро, проходя через вершину добавляет к степени этой вершины двойку).
- 2) Достаточность (справа налево). Пусть степени всех вершин четны и граф связен. Докажем, что граф является эйлеровым. Начнем строить цепь P_1 из произвольной вершины V_i и будем продолжать эту цепь, каждый раз выбирая новое ребро. Если вершина, в которую пришла цепь P_1 , отлична от вершины V_i , то всегда имеется новое ребро для выхода из вершины в силу четности степеней. Если мы вернулись в вершину V_i , то построение цепи P_1 заканчивается циклом, то есть P_1 – цикл. Если цикл P_1 содержит все ребра графа, то это эйлеров цикл, иначе

удаляем из графа G цикл P_1 . Получим граф $G_1 = G \setminus P_1$. Так как граф G и цикл P_1 имеют вершины четных степеней, то граф G_1 содержит только четные степени вершин. В силу связности графа G , графы P_1 и G_1 должны иметь хотя бы одну общую вершину. Для ясности назовем эту вершину V_j (см. Рис. 58).

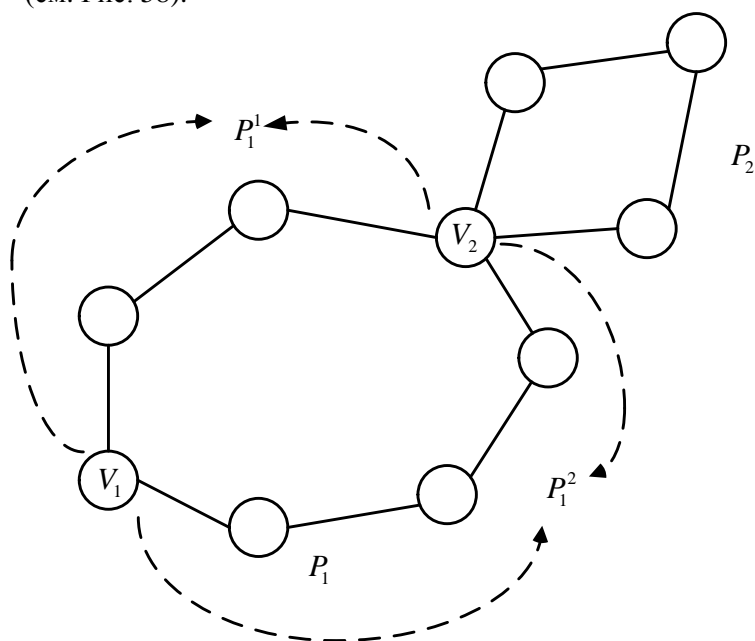


Рис. 58. Иллюстрация к теореме о четности степеней вершин в эйлеровом графе.

Начиная с вершины V_j , строим цикл в графе G_1 аналогично построению цикла P_1 в графе G . Цикл P_1 разбивается между вершинами V_i и V_j на 2 части. Назовем их P_1^1 и P_1^2 , получим новый цикл $P_3 = P_1^1 \cup P_2 \cup P_1^2$.

Если P_3 – не эйлеров цикл, то продолжаем поиск, пока не

обойдем все ребра, так как G – конечный граф, то эйлеров цикл будет построен. ■

Определение. Эйлеровым путем в графе называется произвольный путь, проходящий через каждое ребро графа точно один раз.

Замечание. Эйлеров путь начинается в одной вершине с нечетной степенью и заканчивается в другой вершине нечетной степенью. Эйлеров путь в графе существует тогда и только тогда, когда граф связный и содержит не более, чем две вершины нечетной степени. (Эйлеров путь получается удалением одного ребра из эйлерова цикла.)

Алгоритм поиска эйлерова цикла в графе (пошагово)

Обозначения: $stack$ – рабочий стек (система обслуживания LIFO, Last In First Out);

res – стек, содержащий последовательность вершин эйлерова цикла;

$u(V_i)$ – упорядоченный по нумерации список смежности для вершины V_i ,
 $u(V_i), V_i = \overline{V_1; V_n}, n = |\mathbf{V}|.$

Начало: $G = (\mathbf{V}, \mathbf{E})$ – связный неорграф без вершин нечетной степени. Для каждой вершины $V_i \in \mathbf{V}$ построим список смежности $u(V_i)$, упорядоченный по нумерации.
 $stack := \emptyset$ и $res := \emptyset$.

Шаг 1. Выбираем начальную вершину $V_i \in \mathbf{V}$ и $V_i \rightarrow stack$. (В задаче будет указана начальная вершина или начальная вершина будет выбираться по нумерации.)

Шаг 2. Список смежности для вершины $V_i \in \mathbf{V}$ может быть пуст или содержать некоторые вершины. В зависимости от этого возникает 2 случая:

1 случай $u(V_i) \neq \emptyset$:

$u(V_i) = \{V_j, \dots\}$, выбираем из данного списка

смежности по нумерации первую вершину $V_j \in \mathbf{V}$.

$$u(V_i) := u(V_i) \setminus \{V_j\},$$

$$u(V_j) := u(V_j) \setminus \{V_i\}.$$

Далее для списка смежности вершины V_j возможны два варианта:

Вариант А: $u(V_j) \neq \emptyset \Rightarrow V_j \rightarrow \text{stack}$, $V_i := V_j$ – текущей вершиной становится V_j и возвращаемся к началу шага 2.

Вариант Б: $u(V_j) = \emptyset \Rightarrow V_j \rightarrow \text{res}$, вершина V_i снова становится верхней вершиной стека stack . Далее, возвращаемся к началу шага 2.

2 случай $u(V_i) = \emptyset$

$V_i \rightarrow \text{res}$, и предыдущая вершина из рабочего стека stack до V_i становится верхней вершиной стека. Далее возвращаемся в начале шага 2.

Конец: $\text{stack} = \emptyset \Rightarrow$ эйлеров цикл построен и представлен в res .

Пример 41. Для графа на Рис. 59 найти эйлеров цикл в графе, начиная с вершины V_1 .

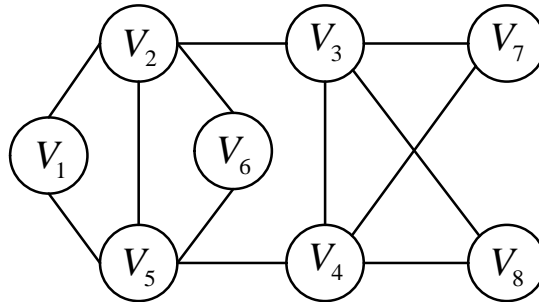


Рис. 59. Исходный граф для поиска эйлерова цикла.

Составим списки смежности для всех вершин

$$u(V_1) = \{V_2, V_5\},$$

$$u(V_2) = \{V_1, V_2, V_5, V_6\},$$

res	V_3	V_7	V_4	V_3	V_2	V_1
res	V_7	V_4	V_3	V_2	V_1	
res	V_4	V_3	V_2	V_1		
res	V_3	V_2	V_1			
res	V_2	V_1				
res	V_1					

Ответ: Эйлеров цикл записывается следующим образом

$$\text{res} = \{V_1, V_2, V_3, V_4, V_7, V_3, V_8, V_4, V_5, V_2, V_6, V_5, V_1\}.$$

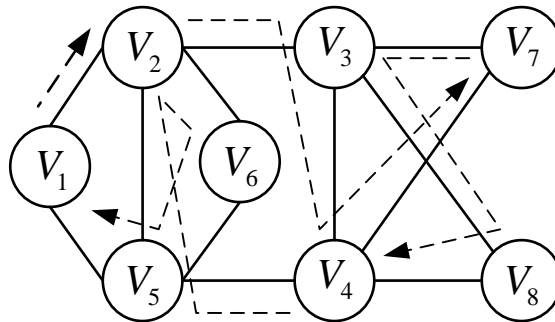


Рис. 60. Эйлеров цикл.

На Рис. 60 изображен проход алгоритма поиска Эйлера цикла, начиная из вершины V_1 , проходим в вершину V_2 и так далее, при этом вершины берем из полученного списка res. Получаем, что мы посетили все вершины и прошли все ребра графа.

Тема 7. Гамильтоновы графы. Сходство и различия гамильтоновых и эйлеровых графов. Достаточные условия существования гамильтоновых циклов. Способы поиска гамильтонова цикла. Алгоритм поиска гамильтонова цикла в графе

Гамильтоновы графы

Определение. Простой цикл называют гамильтоновым, если он включает в себя все вершины связного неорграфа.

Определение. Граф, в котором есть гамильтонов цикл называют гамильтоновым графом (Рис. 61).

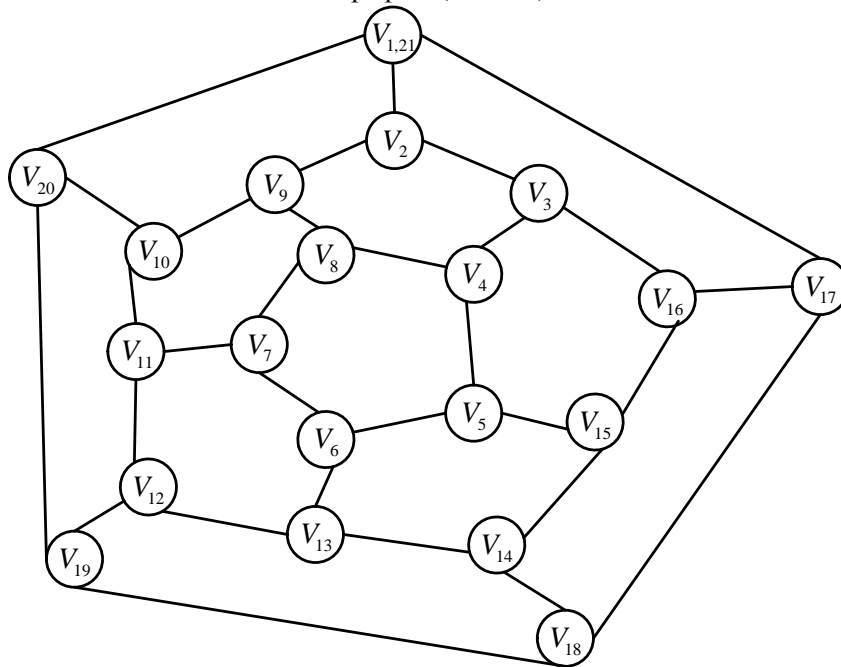


Рис. 61. Граф «Кругосветное путешествие».

История: Слово «гамильтонов» в этих определениях является производным от имени математика Уильяма Роуэна

Гамильтона, предложившего своим друзьям игру по названию «Кругосветное путешествие». Для игры Гамильтон изобразил граф, содержащий 20 вершин, названиями которых служили названия городов. Цель игры – совершить кругосветное путешествие, посетив каждый город один раз, и вернуться домой. Очевидно, что задача сводилась к поиску в графе простого цикла, проходимого через все вершины. На Рис. 61 изображен граф, содержащий гамильтонов цикл. Вершины этого графа пронумерованы в порядке обхода гамильтонова цикла одним из множества способов.

Пример 42. Для графа на Рис. 62 гамильтонова цепь, тогда как гамильтонов цикл не существует. Читателю предоставляется возможность ответить на вопрос, почему не существует гамильтонова цикла, и выписать порядок вершин в гамильтоновой цепи.

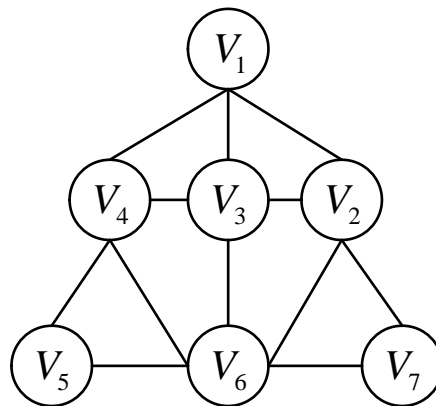


Рис. 62. Гамильтонова цепь.

Пример 43. Для графа, представленного на Рис. 63, не существует ни гамильтонова цикла, ни гамильтоновой цепи. Читателю предоставляется возможность самостоятельно пронумеровать вершины графа и найти причину отсутствия гамильтоновой цепи и цикла.

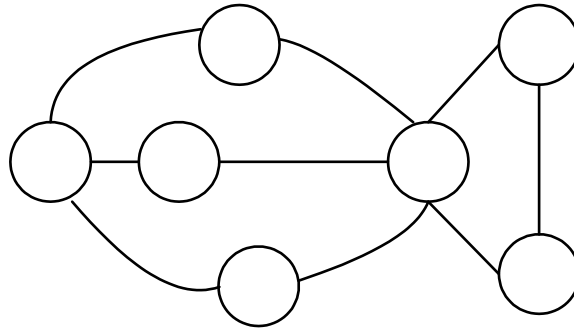


Рис. 63. Граф «Рыбка», не содержащий гамильтоновых цепей и циклов.

Выйдя из любой вершины, нельзя обойти все вершины один раз и вернуться в исходную.

Сходство и различия гамильтоновых графов с эйлеровыми графами

В Табл. 7 представлены сходства и различия гамильтоновых циклов с эйлеровыми циклами.

Табл. 7. Эйлеровы и гамильтоновы циклы.

Эйлеровы циклы	Гамильтоновы циклы
Эйлеров цикл проходит по каждому ребру один раз.	Гамильтонов цикл проходит один раз по всем вершинам.
Эйлеров цикл может проходить через одну вершину несколько раз.	Гамильтонов цикл может не проходить по некоторым ребрам.

Достаточные условия существования гамильтоновых циклов

Для гамильтоновых графов не существует одного необходимого и достаточного условия существования цикла, как у эйлерова графа (четность степеней вершин). Известны лишь несколько достаточных условий существования гамильтоновых

циклов в неорграфах.

(1) Теорема Оре. Если для любой пары несмежных вершин $\{V_i, V_j\}$ графа $G=(\mathbf{V}, \mathbf{E})$ порядка $|\mathbf{V}| \geq 3$ выполняется неравенство $\delta(V_i) + \delta(V_j) \geq |\mathbf{V}|$, то граф G – гамильтонов граф.

(2) Теорема. Если для любой вершины V_i графа $G=(\mathbf{V}, \mathbf{E})$ порядка $|\mathbf{V}| > 3$ выполняется неравенство $\delta(V_i) \geq \frac{|\mathbf{V}|}{2}$, то граф G – гамильтонов граф.

(3) Теорема. Любой 4–связный планарный граф является гамильтоновым.

Замечание к (3). Связность (реберная) определяется как наименьшее количество ребер, удаление которых приводит к несвязному графу.

Способы поиска гамильтонова цикла

1 способ. Наиболее простой, но наиболее трудоемкий. После выбора начальной вершины V_s рассматриваются различные перестановки из множества остальных $(|\mathbf{V}|-1)$ вершин, и для каждой перестановки проверяется наличие гамильтоновой цепи. Если такая цепь найдена, и последняя вершина цепи связана с вершиной V_s , то найден и гамильтонов цикл.

2 способ. Второй способ предлагается рассматривать для орграфов, в случае неорграфов каждое ребро может быть заменено на две дуги в одном и другом направлении.

Алгоритм нахождения гамильтонова цикла

Введем следующие обозначения.

- 1) $P^{(l)}$ – матрица всех маршрутов с l промежуточными вершинами для всех упорядоченных пар вершин графа.

$$P^{(l)} = P' \times P^{(l-1)},$$

$$l = \overline{1, |V| - 1}.$$

$$2) \quad P' = [p'_{i,j}]_{i,j \in \overline{1, |V|}}$$

$$p'_{i,j} = \begin{cases} 0, & \text{если } \langle V_i, V_j \rangle \in \mathbf{E}, \\ V_j, & \text{если } \langle V_i, V_j \rangle \notin \mathbf{E} \end{cases}$$

Индекс j – название вершины.

$$3) \quad P^{(0)} = [p^{(0)}_{i,j}]_{i,j \in \overline{1, |V|}} \text{ получаем из матрицы } P'.$$

$$p^{(0)}_{i,j} = \begin{cases} 0, & \text{если } p'_{i,j} = 0, \\ 1, & \text{если } p'_{i,j} \neq 0. \end{cases}$$

Матрицу всех гамильтоновых цепей графа можно получить, исключив из матрицы $P^{(|V|-2)}$ произведения с повторяющимися сомножителями (считая и конечные вершины цепей). Диагональные элементы матрицы $P^{(|V|-2)}$ после такой обработки описывают все простые циклы графа, содержащие $|V|-1$ вершин, а диагональные элементы матрицы $P^{(|V|-1)}$ дают все простые циклы графа, содержащие $|V|$ вершин, то есть гамильтоновы циклы. Причем любой диагональный элемент отражает все циклы.

Упрощение. Вместо матрицы $P^{(|V|-1)}$ достаточно сформировать только один из ее столбцов, соответствующий начальной вершине V_s . Искомый результат содержится в строке s найденного столбца.

- 4) $P_s^{(l)}$ – столбец s матрицы $P^{(l)}$.
- 5) $P_s^{(l)} = P' \times P_s^{(l-1)}$, где $l = 1, 2, \dots, (|V|-1)$.
- 6) $P_s^{(0)}$ – столбец номер s матрицы $P^{(0)}$.

После введенных обозначений можно перейти к самому алгоритму нахождения гамильтонова цикла.

Алгоритм поиска гамильтонова цикла в графе

Дано. $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – связный орграф. Определена начальная вершина V_s для построения гамильтонова цикла.

Шаг 1. Составляется матрица P' и столбец $P_s^{(0)}$.

Шаг 2. $P_s^{(l)} = P' \times P_s^{(l-1)}$, $l = \overline{1; |V| - 1}$,

в столбце $P_s^{(l)}$ обнуляются элементы:

- 1) в столбце $P_s^{(l)}$, если «произведение вершин» в строке содержит вершину, равную метке строки, $l = \overline{1; |V| - 1}$;
- 2) все элементы в s -ой строке столбца $P_s^{(l)}$ $l = \overline{1; |V| - 2}$;
- 3) в столбце $P_s^{(l)}$ произведение, содержащее одинаковые множители $l = \overline{1; |V| - 1}$.

Шаг 3. При $l = |V| - 1$ в s -той строке получим количество гамильтоновых циклов и последовательности вершин в цикле.

Конец алгоритма. Перечисляем количество циклов в графе G и последовательности вершин в каждом цикле.

Пример 44. Для графа на Рис. 64 найти количество гамильтоновых циклов из вершины V_1 и указать их.

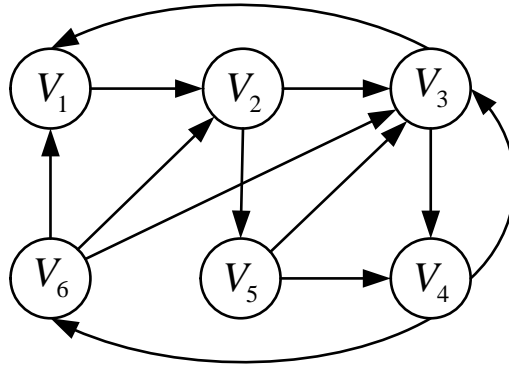


Рис. 64. Граф с гамильтоновыми циклами.

$$P' = \begin{pmatrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \\ V_1 & 0 & V_2 & 0 & 0 & 0 & 0 \\ V_2 & 0 & 0 & V_3 & 0 & V_5 & 0 \\ V_3 & V_1 & 0 & 0 & V_4 & 0 & 0 \\ V_4 & 0 & 0 & V_3 & 0 & 0 & V_6 \\ V_5 & 0 & 0 & V_3 & V_4 & 0 & 0 \\ V_6 & V_1 & V_2 & V_3 & 0 & 0 & 0 \end{pmatrix}, P_{V_1}^{(0)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

В графе шесть вершин, следовательно, будет пять итераций.

$$1) \quad P' \times P_{V_1}^{(0)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} 0 \\ V_3 \\ 0 \\ V_3 + V_6 \\ V_3 \\ V_3 \end{bmatrix} = P_{V_1}^{(1)}.$$

$$2) \quad P' \times P_{V_1}^{(1)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} \frac{V_2 V_3}{V_5 V_3} \\ \frac{V_4 V_3 + V_4 V_6}{V_6 V_3} \\ V_4 V_3 + V_4 V_6 \\ V_2 V_3 \end{bmatrix} \Rightarrow P_{V_1}^{(2)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} 0 \\ V_5 V_3 \\ V_4 V_6 \\ V_6 V_3 \\ V_4 V_3 + V_4 V_6 \\ V_2 V_3 \end{bmatrix}.$$

$$3) \quad P' \times P_{V_1}^{(2)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} \frac{V_2 V_5 V_3}{V_3 V_4 V_6 + V_5 V_4 V_3 + V_5 V_4 V_6} \\ \frac{V_4 V_6 V_3}{V_3 V_4 V_6 + V_6 V_2 V_3} \\ V_3 V_4 V_6 + V_4 V_6 V_3 \\ V_2 V_5 V_3 + \frac{V_3 V_4 V_6}{V_3 V_4 V_6} \end{bmatrix} \Rightarrow P_{V_1}^{(3)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} 0 \\ V_3 V_4 V_6 + V_5 V_4 V_3 + V_5 V_4 V_6 \\ 0 \\ V_6 V_2 V_3 \\ V_3 V_4 V_6 + V_4 V_6 V_3 \\ V_2 V_5 V_3 \end{bmatrix}.$$

$$4) \quad P' \times P_{V_1}^{(3)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} \frac{V_2 V_3 V_4 V_6 + V_2 V_5 V_4 V_3 + V_2 V_5 V_4 V_6}{V_5 V_3 V_4 V_6 + V_5 V_4 V_6 V_3} \\ \frac{V_4 V_6 V_2 V_3}{V_6 V_2 V_5 V_3} \\ V_4 V_6 V_2 V_3 \\ \frac{V_2 V_3 V_4 V_6 + V_2 V_5 V_4 V_3 + V_2 V_5 V_4 V_6}{V_5 V_3 V_4 V_6 + V_5 V_4 V_6 V_3} \end{bmatrix} \Rightarrow$$

$$\begin{aligned}
& P_{V_1}^{(4)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} 0 \\ V_5 V_3 V_4 V_6 + V_5 V_4 V_6 V_3 \\ 0 \\ V_6 V_2 V_5 V_3 \\ V_4 V_6 V_2 V_3 \\ V_2 V_5 V_4 V_3 \end{bmatrix} \\
5) \quad P' \times P_{V_1}^{(4)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} V_2 V_5 V_3 V_4 V_6 + V_2 V_5 V_4 V_6 V_3 \\ \frac{V_5 V_4 V_6 V_2 V_3}{V_4 V_6 V_2 V_5 V_3} \\ \frac{V_4 V_6 V_2 V_5 V_3}{V_6 V_2 V_5 V_4 V_3} \\ \frac{V_6 V_2 V_5 V_4 V_3}{V_4 V_6 V_2 V_5 V_3} \\ \frac{V_4 V_6 V_2 V_5 V_3}{V_2 V_5 V_3 V_4 V_6 + V_2 V_5 V_4 V_6 V_3} \\ V_2 V_5 V_3 V_4 V_6 + V_2 V_5 V_4 V_6 V_3 \end{bmatrix} \Rightarrow \\
& P_{V_1}^{(5)} = \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{matrix} \begin{bmatrix} V_2 V_5 V_3 V_4 V_6 + V_2 V_5 V_4 V_6 V_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow
\end{aligned}$$

Получаем два цикла: 1) $V_1 V_2 V_5 V_3 V_4 V_6 V_1$ и 2) $V_1 V_2 V_5 V_4 V_6 V_3 V_1$.
Далее в качестве проверки получаем дуги, которые входят в цикл, на Рис. 65 и Рис. 66 они представлены пунктирными линиями.

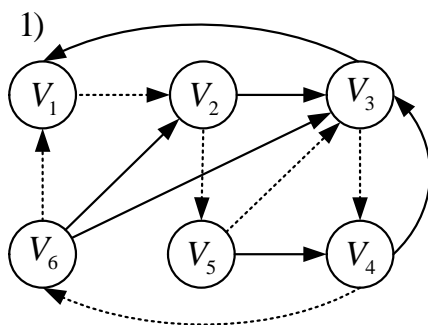


Рис. 65. Первый гамильтонов
цикл.

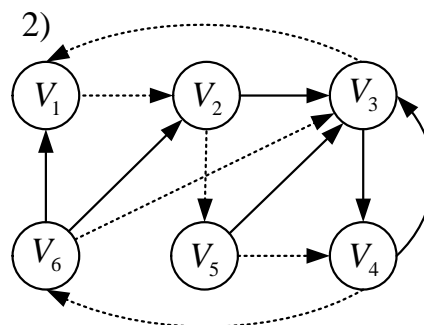


Рис. 66. Второй гамильтонов
цикл.

Ответ: Два гамильтонова цикла (Рис. 65 и Рис. 66).

1) $V_1V_2V_5V_3V_4V_6V_1$ и 2) $V_1V_2V_5V_4V_6V_3V_1$.

**Тема 9. Поиск расстояния между всеми парами вершин.
Алгоритм Уоршалла-Флойда**

Для решения задачи определения расстояния между всеми парами вершин можно использовать $n(n-1)$ раз алгоритм Дейкстры для каждой вершины. Однако, сложность такой процедуры будет составлять $O(n^4)$. Но существуют более эффективные алгоритмы, в частности, алгоритм Уоршалла-Флойда. Любое расстояние между двумя вершинами сравнивается с длиной маршрута через все остальные вершины.

Алгоритм Уоршалла-Флойда

Начало. Рассмотрим орграф $G = \langle \mathbf{V}, \mathbf{E} \rangle$, где $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, $|\mathbf{V}| = n$.

Пусть матрица $D^{(m)} = [d_{i,j}^{(m)}]$ и ее элементы $d_{ij}^{(m)} = d^{(m)}(V_i, V_j)$, $i, j = \overline{1, n}$ – длина кратчайшего из путей из V_i в V_j с промежуточными вершинами из множества $\{V_1, V_2, \dots, V_m\}$, $m \leq n$.

Шаг 1. Построение матрицы $D^{(0)}$.

$$D^{(0)} := [d_{ij}^{(0)}]_{i,j=\overline{1,n}} = \begin{cases} \min(w_{ij}), & \text{если } \langle V_i, V_j \rangle \in \mathbf{E}, \\ \infty, & \text{если } \langle V_i, V_j \rangle \notin \mathbf{E}. \end{cases}$$

Шаг 2. $m := m + 1$. Построение матрицы $D^{(m)}$, $m = \overline{1, n}$:

$$d_{ij}^{(m)} := \min(d_{ij}^{(m-1)}, d_{im}^{(m-1)} + d_{mj}^{(m-1)}).$$

- 1) Если $m < n$, то вернуться к началу шага 2.
- 2) Если $m = n$, то $D^{(n)}$ – и есть матрица кратчайших путей между всеми парами вершин.

Конец алгоритма. $D^{(n)}$ – матрица кратчайших путей в графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$.

Сложность алгоритма Уоршалла-Флойда $O(n^3)$.

Пример 45. Найти минимальное расстояние между всеми парами вершин для графа на Рис. 67 по алгоритму Уоршалла-Флойда.

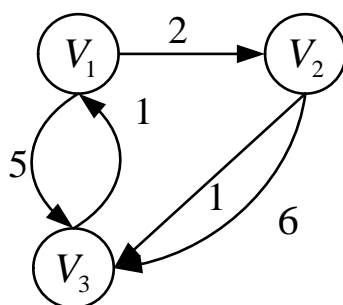


Рис. 67. Исходный граф.

Матрица $D^{(0)}$:

$D^{(0)}$	V_1	V_2	V_3
V_1	∞	2	5
V_2	∞	∞	1
V_3	1	∞	∞

Для решения задачи применяем алгоритм Уоршалла-Флойда.

1) $d_{i,j}^{(1)} := \min(d_{i,j}^{(0)}, d_{i,1}^{(0)} + d_{1,j}^{(0)})$, $i, j = \overline{1,3}$

$D^{(1)}$	V_1	V_2	V_3
V_1	∞	2	5
V_2	∞	∞	1
V_3	1	3	6

, первая строка и первый столбец не меняются

в силу определения матрицы $D^{(1)}$.

2) $d_{i,j}^{(2)} := \min(d_{i,j}^{(1)}, d_{i,2}^{(1)} + d_{2,j}^{(1)})$, $i, j = \overline{1,3}$,

$$D^{(2)} \begin{array}{c|c|c|c} & V_1 & V_2 & V_3 \\ \hline V_1 & \infty & 2 & 3 \\ \hline V_2 & \infty & \infty & 1 \\ \hline V_3 & 1 & 3 & 4 \end{array} \text{ . вторая строка и второй столбец не меняются}$$

в силу определения матрицы $D^{(2)}$.

$$3) \quad d_{i,j}^{(3)} := \min(d_{i,j}^{(2)}, d_{i,3}^{(2)} + d_{3,j}^{(2)}), \quad i, j = \overline{1,3},$$

$$D^{(3)} \begin{array}{c|c|c|c} & V_1 & V_2 & V_3 \\ \hline V_1 & 4 & 2 & 3 \\ \hline V_2 & 2 & 4 & 1 \\ \hline V_3 & 1 & 3 & 4 \end{array} \text{ третья строка и третий столбец не меняются в}$$

силу определения матрицы $D^{(3)}$.

Элементы матрицы $D^{(3)}$ показывают кратчайшие расстояния между всеми парами вершин. Например, расстояние от V_3 к V_2 , равно трем, получается переходом из вершины V_3 в V_1 , вес этой дуги равен 1, и из V_1 в V_2 , вес дуги 2, суммируя, получаем 3.

$$\underline{\text{Ответ.}} \quad D^{(3)} \begin{array}{c|c|c|c} & V_1 & V_2 & V_3 \\ \hline V_1 & 4 & 2 & 3 \\ \hline V_2 & 2 & 4 & 1 \\ \hline V_3 & 1 & 3 & 4 \end{array} \text{ .}$$

Тема 10. Задача построения транзитивного замыкания бинарного отношения. Алгоритм построения транзитивного замыкания бинарного отношения

Задача построения транзитивного замыкания бинарного отношения

С задачей определения кратчайших путей в графе тесно связана задача транзитивного замыкания бинарного отношения.

Определение. Рассмотрим орграф $G = \langle \mathbf{V}, \mathbf{E} \rangle$, где $|\mathbf{V}| = n$. Под бинарным отношением на множестве \mathbf{V} мы будем понимать произвольное подмножество $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ (множество дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$). Бинарное отношение \mathbf{E} можно однозначно представить графом $G = \langle \mathbf{V}, \mathbf{E} \rangle$.

Определение. Бинарное отношение на графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$ является транзитивным при выполнении условия: если $\langle x, y \rangle \in \mathbf{E}$ и $\langle y, z \rangle \in \mathbf{E}$, то $\langle x, z \rangle \in \mathbf{E}$ для произвольных вершин $x, y, z \in \mathbf{V}$.

Определение. $\mathbf{E}^* = \{ \langle x, y \rangle : \text{в } G = \langle \mathbf{V}, \mathbf{E} \rangle \exists \text{ путь нулевой длины из } x \text{ в } y \}$, \mathbf{E}^* – транзитивное отношение на множестве \mathbf{V} и $\mathbf{E} \subseteq \mathbf{E}^*$. Отношение \mathbf{E}^* называется транзитивным замыканием.

Алгоритм построения транзитивного замыкания

Алгоритм вычисления транзитивного замыкания имеет сходство с алгоритмом Уоршалла-Флойда.

Введем следующие обозначения.

$$D^{(0)} = [d_{i,j}^{(0)}], \quad i, j = \overline{1, n}$$
$$d_{i,j}^{(0)} := \begin{cases} 1, & \text{если } \langle V_i, V_j \rangle \in \mathbf{E} \\ 0, & \text{если } \langle V_i, V_j \rangle \notin \mathbf{E} \end{cases} \quad (10.1)$$

Начало. Граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$, $|\mathbf{V}| = n$.

Шаг 1. Построение начальной матрицы $D^{(0)}$, используя формулу (10.1).

Шаг 2. Построение матрицы $D^{(m)}$, $m = \overline{1, n}$:

$$D^{(m)} = [d_{i,j}^{(m)}], \quad i, j = \overline{1, n},$$

$$d_{i,j}^{(m)} = \max(d_{i,j}^{(m-1)}; d_{i,m}^{(m-1)} \cdot d_{m,j}^{(m-1)}).$$

А) если $m < n \Rightarrow$ возврат к началу шага 2,

Б) если $m = n$, то $D^{(n)}$ – является матрицей связности (достижимости) графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$. Переходим к шагу 3.

Шаг 3. Определяем транзитивное замыкание по матрице связности следующим образом:

$$D^{(n)} = [d_{i,j}^n], \quad d_{i,j}^n = \begin{cases} 1, & \text{если } \langle V_i, V_j \rangle \in \mathbf{E}^*, \\ 0, & \text{если } \langle V_i, V_j \rangle \notin \mathbf{E}^*. \end{cases}$$

Конец алгоритма. По матрице связности строится транзитивное замыкание.

Пример 46. Найти матрицу связности и построить транзитивное замыкание для графа, представленного на Рис. 68.

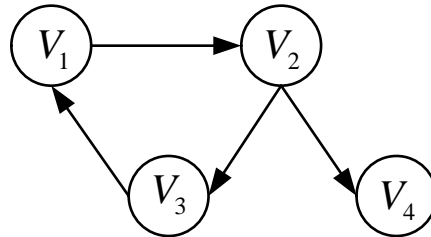


Рис. 68. Исходный граф.

Применяя алгоритм построения транзитивного замыкания находим последовательно все матрицы.

$D^{(0)}$	V_1	V_2	V_3	V_4
V_1	0	1	0	0
V_2	0	0	1	1
V_3	1	0	0	0
V_4	0	0	0	0

в следующей матрице сразу проставим единицы, так как это максимум, а нули могут как поменяться, так и остаться.

$$1) D^{(1)} = [d_{i,j}^{(1)}], \quad i, j = \overline{1,4}; \quad d_{i,j}^{(1)} = \max(d_{i,j}^{(0)}; d_{i,1}^{(0)} \times d_{1,j}^{(0)}).$$

Получаем:

$D^{(1)}$	V_1	V_2	V_3	V_4
V_1	0	1	0	0
V_2	0	0	1	1
V_3	1	1	0	0
V_4	0	0	0	0

$$2) D^{(2)} = [d_{i,j}^{(2)}], \quad i, j = \overline{1,4}; \quad d_{i,j}^{(2)} = \max(d_{i,j}^{(1)}; d_{i,2}^{(1)} \times d_{2,j}^{(1)}).$$

$D^{(2)}$	V_1	V_2	V_3	V_4
V_1	0	1	1	1
V_2	0	0	1	1
V_3	1	1	1	1
V_4	0	0	0	0

$$3) D^{(3)} = [d_{i,j}^{(3)}], \quad i, j = \overline{1,4}; \quad d_{i,j}^{(3)} = \max(d_{i,j}^{(2)}; d_{i,3}^{(2)} \times d_{3,j}^{(2)}).$$

$D^{(3)}$	V_1	V_2	V_3	V_4
V_1	1	1	1	1
V_2	1	1	1	1
V_3	1	1	1	1
V_4	0	0	0	0

$$4) D^{(4)} = [d_{i,j}^{(4)}], \quad i, j = \overline{1,4}; \quad d_{i,j}^{(4)} = \max(d_{i,j}^{(3)}; d_{i,4}^{(3)} \times d_{4,j}^{(3)}).$$

$D^{(4)}$	V_1	V_2	V_3	V_4
V_1	1	1	1	1
V_2	1	1	1	1
V_3	1	1	1	1
V_4	0	0	0	0

Построим транзитивное замыкание по матрице связности:

$$\mathbf{E}^* = \left\{ \begin{array}{l} \langle V_1, V_1 \rangle, \langle V_1, V_2 \rangle, \langle V_1, V_3 \rangle, \langle V_1, V_4 \rangle, \\ \langle V_2, V_1 \rangle, \langle V_2, V_2 \rangle, \langle V_2, V_3 \rangle, \langle V_2, V_4 \rangle, \\ \langle V_3, V_1 \rangle, \langle V_3, V_2 \rangle, \langle V_3, V_3 \rangle, \langle V_3, V_4 \rangle \end{array} \right\}.$$

Построим результирующий граф Рис. 69:

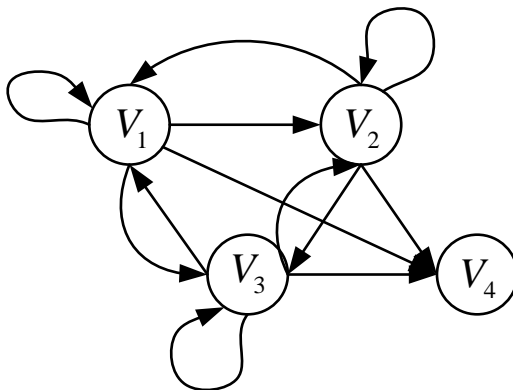


Рис. 69. Транзитивное замыкание для графа на Рис. 68.

Тема 11. Потoki. Условия существования потока. Увеличивающая цепь. Алгоритм поиска увеличивающей цепи. Увеличение потока вдоль найденной цепи по правилам

Потоки

Деятельность современного общества тесно связана с разного рода сетями, на пример, транспортные сети, коммуникации, распределение товаров. Управление потоками в транспортных сетях имеет важное значение. Читателю предлагается ознакомиться с основными задачами, встречающимися в этой области, решаемые с помощью теории графов.

Поток определяет способ пересылки некоторых объектов из одного пункта в другой. Потoki могут возникнуть при транспортировке товаров, при передвижении людей.

Применительно к графам поток задает способ пересылки некоторых объектов (единиц потока) из одной вершины графа (из источника) в другую вершину (сток) по дугам в направлении их ориентации. Максимальное число единиц потока, которые могут проходить по дуге называют пропускной способностью дуги. Кроме пропускной способности дуг в некоторых задачах задают стоимость прохождения единицы потока по дугам.

Пример графа с заданным потоком величины 7 представлен на

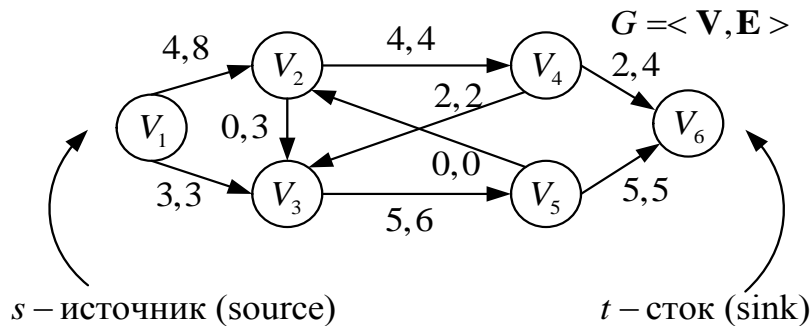


Рис. 70.

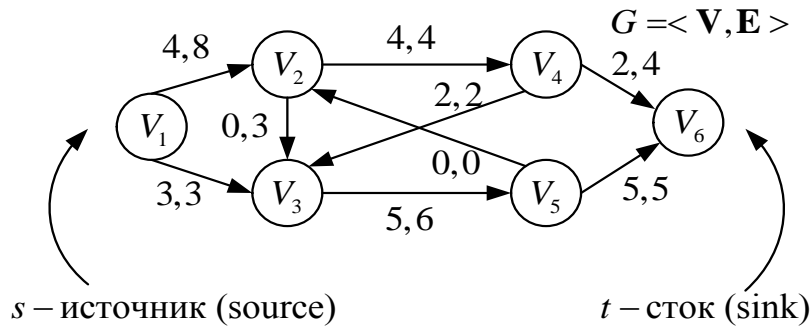


Рис. 70. Пример потока в графе.

Определение. Максимальное число единиц потока, которые могут проходить по дуге, называется пропускной способностью дуги. Обозначается через $c(V_i, V_j)$ ¹¹. (См. рис. 11.2)

Введем следующее обозначение. $f(V_i, V_j)$ ¹² – число единиц потока, проходящих по дуге $\langle V_i, V_j \rangle$ (Рис. 71).

Отметим, что число единиц потока не может превышать пропускную способность: $f(V_i, V_j) \leq c(V_i, V_j)$.

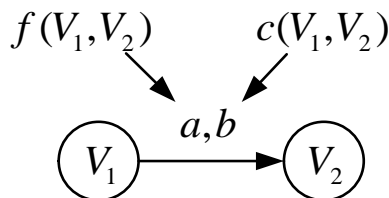


Рис. 71. Обозначение на дуге.

Условия существования потока

Пусть дан $G = \langle \mathbf{V}, \mathbf{E} \rangle$, для каждой дуги $\langle V_i, V_j \rangle \in \mathbf{E}$ существует $f(V_i, V_j)$ и $c(V_i, V_j)$, $V_s \in \mathbf{V}$ – источник и $V_t \in \mathbf{V}$ –

¹¹ c от англ. capacity – пропускная способность.

¹² f от англ. flow – поток.

сток.

$$1) \quad \forall \langle V_i, V_j \rangle \in \mathbf{E}; \quad 0 \leq f(V_i, V_j) \leq c(V_i, V_j) \quad (11.1)$$

Число единиц потока, проходящих по дуге $\langle V_i, V_j \rangle$

ограничено пропускной способностью этой дуги.

$$2) \quad \forall V_i \in \mathbf{V}, \quad V_i \neq V_S, \quad V_i \neq V_T:$$

$$\sum_{V_j \in \mathbf{V}} f(V_i, V_j) = \sum_{V_j \in \mathbf{V}} f(V_j, V_i). \quad (11.2)$$

Условие (11.2) запрещает утечку потока в промежуточных вершинах, число приходящих единиц потока в вершину должно быть равно числу выходящих из этой вершины единиц потока.

$$3) \quad \sum_{V_j \in \mathbf{V}} f(V_S, V_j) - \sum_{V_j \in \mathbf{V}} f(V_j, V_S) = K = \sum_{V_j \in \mathbf{V}} f(V_j, V_T) - \sum_{V_j \in \mathbf{V}} f(V_T, V_j)$$

Распишем каждую сумму в этом уравнении.

$\sum_{V_j \in \mathbf{V}} f(V_S, V_j)$ – количество единиц потока, выходящего из

источника V_S .

$\sum_{V_j \in \mathbf{V}} f(V_j, V_S)$ – количество единиц потока, входящего в источник

V_S .

$\sum_{V_j \in \mathbf{V}} f(V_j, V_T)$ – количество единиц потока, входящего в сток V_T .

$\sum_{V_j \in \mathbf{V}} f(V_T, V_j)$ – количество единиц потока, выходящего из

стока V_T .

Из источника V_S выходит всего (суммарно) K единиц потока, и в сток V_T входит всего (суммарно) K единиц потока.

Условие 3) задает баланс единиц потока, выходящего из источника и входящего в сток.

Если условия 1)–3) соблюдаются, то говорят, что в графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$ **задан поток величины K** (то есть из источника в сток передается K единиц потока).

Увеличивающая цепь

Пусть в графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$, $|\mathbf{V}| = n$, существует поток из вершины $V_s \in \mathbf{V}$ в вершину $V_t \in \mathbf{V}$. Решая транспортные задачи, часто возникает вопрос возможности увеличения потока из вершины V_s в вершину V_t . Для решения подобной задачи предлагается алгоритм поиска увеличивающей цепи. Для перехода к алгоритму приведем дополнительные определения и обозначения.

Определение. Увеличивающей цепью называется цепь, по которой могут быть переданы дополнительные единицы потока.

Введем следующие обозначения.

- 1) **N** (neutral) – группа дуг графа, в которых поток не может изменяться (дуги с нулевой пропускной способностью или дуги, в которых по каким-либо причинам, например техническим, число проходящих единиц потока должно быть постоянным).
- 2) **I** (increase) – дуги, в которых поток может увеличиваться, то есть дуги $\langle V_i, V_j \rangle \in \mathbf{E}$, в которых число проходящих единиц потока $f(V_i, V_j)$ меньше пропускной способности $c(V_i, V_j)$.
- 3) **R** (reduce) – дуги, в которых поток может уменьшаться, то есть дуги $\langle V_i, V_j \rangle \in \mathbf{E}$, в которых число проходящих единиц потока $f(V_i, V_j)$ больше нуля.

Алгоритм поиска увеличивающей цепи в орграфе

Начало. В орграфе $G = \langle \mathbf{V}, \mathbf{E} \rangle$, $|\mathbf{V}| = n$, существует поток, и соблюдены условия существования потока 1)–3).

Шаг 1. Разбить множество дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$ на группы: **N, I, R** (дуги в группах **I, R** могут повторяться).

Шаг 2. 1) Исключить из рассмотрения дуги группы **N**. 2)

окрасить вершину V_s .

Шаг 3. Окрашивать дуги и вершины в соответствии с правилами (правила представлены после алгоритма) до тех пор, пока 1) будет окрашена вершина V_T , 2) окраска новых вершин и дуг станет невозможной.

Шаг 4. 1) Если вершина V_T окрашена, то для цепи выбираются окрашенные дуги, соединяющие вершины V_s и V_T . Такая цепь называется увеличивающей. 2) В обратном случае увеличивающей цепи в графе нет.

Конец алгоритма.

Правила окрашивания дуг $\langle V_i, V_j \rangle$, $\langle V_j, V_i \rangle$ и вершины V_j при уже окрашенной вершине V_i

- 1) Если дуга $\langle V_i, V_j \rangle \in \mathbf{I}$, то дуга $\langle V_i, V_j \rangle$ и вершина V_j окрашиваются. Дуга $\langle V_i, V_j \rangle$ называется прямой дугой.
- 2) Если дуга $\langle V_j, V_i \rangle \in \mathbf{R}$, то окрашивается дуга $\langle V_j, V_i \rangle$ и вершина V_j . Дуга $\langle V_j, V_i \rangle$ называется обратной дугой.
- 3) В остальных случаях дуга $\langle V_i, V_j \rangle$ и вершина V_j не окрашиваются.

Пример 47. Найти увеличивающую цепь для графа, представленного на Рис. 68.

Начало. Граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$, $|\mathbf{V}| = n = 6$.

Шаг 1. Разбиваем множество дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$ на группы: $\mathbf{N}, \mathbf{I}, \mathbf{R}$ (дуги в группах \mathbf{I}, \mathbf{R} могут повторяться).

$$\mathbf{N} = \{ \langle V_5, V_2 \rangle \},$$

$$\mathbf{I} = \{ \langle V_1, V_2 \rangle, \langle V_2, V_3 \rangle, \langle V_3, V_5 \rangle, \langle V_4, V_6 \rangle \},$$

$$\mathbf{R} = \{ \langle V_1, V_3 \rangle, \langle V_2, V_4 \rangle, \langle V_1, V_2 \rangle, \langle V_3, V_5 \rangle, \langle V_4, V_3 \rangle, \langle V_4, V_6 \rangle, \langle V_5, V_6 \rangle \}.$$

Шаг 2. Закрашиваем вершины и дуги. Исключаем дугу $\langle V_5, V_2 \rangle$ и закрашиваем начальную вершину $V_s = V_1$. Результат представлен на Рис. 72.

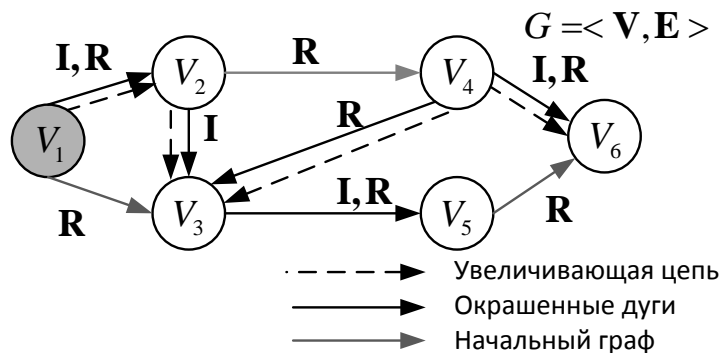


Рис. 72. Окрашивание вершин и дуг.

Шаг 3. По правилу 1.

- 1) Дуга $\langle V_1, V_2 \rangle$ и вершина V_2 окрашиваются.
- 2) Дуга $\langle V_2, V_3 \rangle$ и вершина V_3 окрашиваются.
- 3) Дуга $\langle V_3, V_5 \rangle$ и вершина V_5 окрашиваются.

По правилу 2.

- 4) Дуга $\langle V_1, V_3 \rangle$ не окрашивается, так как из вершины V_1 дуга $\langle V_1, V_3 \rangle$ может только уменьшить поток.
- 5) Дуга $\langle V_4, V_3 \rangle$ и вершина V_4 окрашиваются.

По правилу 1.

- 6) Дуга $\langle V_4, V_6 \rangle$ и вершина V_6 окрашиваются.

Шаг 4. Вершина $V_T = V_6$ окрашена, следовательно, существует увеличивающая цепь от вершины $V_S = V_1$ до $V_T = V_6$.

Получаем следующую цепь: $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_6$.

Конец алгоритма.

Ответ. Увеличивающая цепь для заданного графа – $E' = \{ \langle V_1, V_2 \rangle, \langle V_2, V_3 \rangle, \langle V_4, V_3 \rangle, \langle V_4, V_6 \rangle \}$.

Увеличение потока вдоль увеличивающей цепи

Увеличение потока вдоль найденной увеличивающей цепи находится по следующим правилам.

- 1) Определяется величина $t = \min \{ \gamma(V_i, V_j) \} \quad \forall \gamma(V_i, V_j) \in E'$
где $\gamma(V_i, V_j)$ – разность на дуге $\langle V_i, V_j \rangle$.
Если $\langle V_i, V_j \rangle$ – прямая дуга (окрашена по правилу 1),
то $\gamma(V_i, V_j) = c(V_i, V_j) - f(V_i, V_j)$.
Если $\langle V_i, V_j \rangle$ – обратная дуга (окрашена по правилу 2),
то $\gamma(V_i, V_j) = f(V_i, V_j)$.
- 2) С помощью величины t из предыдущего пункта, изменяем поток.
 - а) Для каждой прямой дуги $\langle V_i, V_j \rangle \in E'$
 $f(V_i, V_j) := f(V_i, V_j) + t$, поток увеличивается на t единиц.
 - б) Для каждой обратной дуги $\langle V_i, V_j \rangle \in E'$
 $f(V_i, V_j) := f(V_i, V_j) - t$, поток уменьшается на t единиц.

Вернемся к примеру и вычислим для графа на Рис. 72 величину t .

- 1) $\langle V_1, V_2 \rangle$ – прямая.
 $\langle V_2, V_3 \rangle$ – прямая.
 $\langle V_4, V_3 \rangle$ – обратная.
 $\langle V_4, V_6 \rangle$ – прямая.
$$t = \min \{ \gamma(V_1, V_2); \gamma(V_2, V_3); \gamma(V_4, V_3); \gamma(V_4, V_6) \} =$$
$$= \min \left\{ \begin{array}{l} c(V_1, V_2) - f(V_1, V_2); c(V_2, V_3) - f(V_2, V_3); \\ f(V_4, V_3); c(V_4, V_6) - f(V_4, V_6) \end{array} \right\} =$$
$$= \min \{ (8 - 4); (3 - 0); 2; (4 - 2) \} = \min \{ 4; 3; 2; 2 \} = 2.$$

Получаем $t = 2$.
- 2) $f(V_1, V_2) := 4 + 2 = 6$,

$$f(V_2, V_3) := 2 + 0 = 2,$$

$$f(V_4, V_3) := 2 - 2 = 0,$$

$$f(V_4, V_6) := 2 + 2 = 4.$$

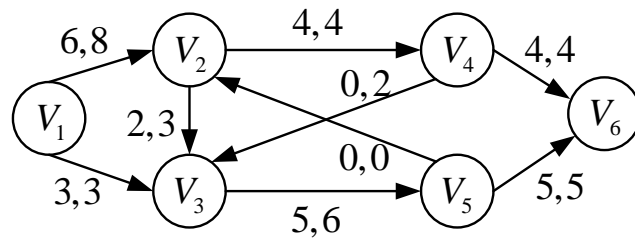


Рис. 73. Результирующий граф.

На Рис. 73 суммарный поток из вершины $V_s = V_1$ в вершину $V_T = V_6$ увеличился на 2 единицы.

Тема 12. Потoki. Поиск максимального потока. Поиск потока минимальной стоимости

Потоки

В прошлой теме исследовались потоки в графах. Такие задачи как поиск увеличивающей цепи в графах и увеличение потока являются дополнительными к более сложным задачам. Продолжая работать с потоками в графах, разберем следующие две задачи:

- 1) поиск максимального потока в графе,
- 2) поиск потока минимальной стоимости в графе.

Поиск максимального потока

Задача о максимальном потоке состоит в поиске способа пересылки максимального количества единиц потока из источника в сток.

Алгоритм поиска максимального потока

Начало. Дан граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$, $|\mathbf{V}| = n$. Заданы источник V_S и сток V_T .

Шаг 1. Проверить, существует ли в графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$ поток из источника V_S в сток V_T , удовлетворяющий условиям существования потока 1)–3).

Шаг 2. Применить к графу $G = \langle \mathbf{V}, \mathbf{E} \rangle$ алгоритм поиска увеличивающей цепи.

Шаг 3. Если увеличивающая цепь найдена, то увеличить поток вдоль найденной цепи по правилам увеличения потока и вернуться к началу шага 2. Иначе, алгоритм закончить, максимальный поток найден.

$$K = \sum_{V_j \in \mathbf{V}} f(V_S, V_j) = \sum_{V_j \in \mathbf{V}} f(V_j, V_T).$$

Конец алгоритма. Максимальный поток найден.

Пример 48. Найти для графа, представленного на Рис. 74, максимальный поток.

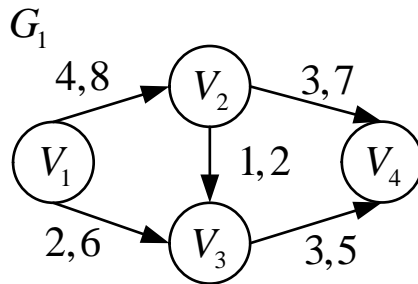


Рис. 74. Исходный граф.

Начальный поток 6 единиц.

1) Первая увеличивающая цепь:

$E_1 = \{ \langle V_1, V_2 \rangle, \langle V_2, V_4 \rangle \}$, $t = 4$ (так как дуги являются прямыми, $t = c(V_i, V_j) - f(V_i, V_j)$). Результат изображен на

Рис. 75.

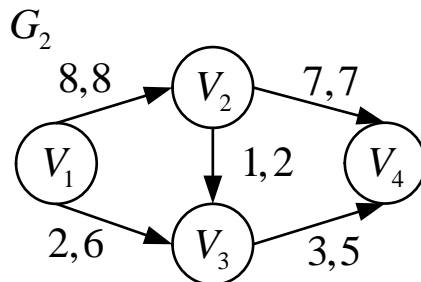


Рис. 75. Граф после первого увеличения потока.

2) Вторая увеличивающая цепь:

$E_2 = \{ \langle V_1, V_3 \rangle, \langle V_3, V_4 \rangle \}$, $t = 2$. Результат представлен на Рис. 76.

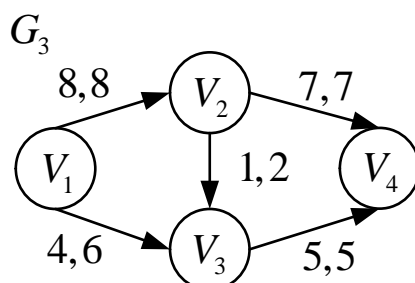


Рис. 76. Граф после второго увеличения потока.

Поток в графе еще раз увеличить невозможно, следовательно, $K_{\max} = 12$ единиц потока.

Ответ. Максимальный поток равен 12 единицам, $K_{\max} = 12$.

Поиск потока минимальной стоимости

Введем следующие обозначения.

- 1) k – число единиц потока из источника V_S в сток V_T с минимальной стоимостью.
- 2) $a(V_i, V_j)$ – стоимость прохождения единицы потока по дуге $\langle V_i, V_j \rangle$.

В задаче о потоке минимальной стоимости требуется переслать фиксированное число k единиц потока из V_S в сток V_T с минимальной стоимостью, то есть найти значение $\min \left\{ \sum_{\langle V_i, V_j \rangle \in E} a(V_i, V_j) \times f(V_i, V_j) \right\}$ при условиях существования потока 1)–3).

- 3) Начальный поток – $f(V_i, V_j)$, стоимость – $a(V_i, V_j)$, пропускная способность – $c(V_i, V_j)$ (см. Рис. 77).

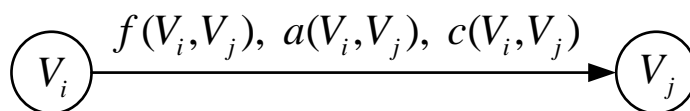


Рис. 77. Обозначения на дугах графа.

Для дуги $\langle V_i, V_j \rangle$ здесь и далее $f, a, c \in \mathbf{Z}$ ¹³.

Идея алгоритма поиска потока минимальной стоимости

Подробный алгоритм не представлен в данном пособии, так как существует множество различных вариаций алгоритма через метки вершин, окрашивание ребер, нумерацию вершин и ребер и т.д. Алгоритм зависит даже от типа задачи, например, с несколькими источниками и стоками. Ниже приведена идея упрощенного алгоритма.

Дано. В графе $G = \langle \mathbf{V}, \mathbf{E} \rangle$ задан нулевой поток, то есть $f(V_i, V_j) = 0, \forall \langle V_i, V_j \rangle \in \mathbf{E}$.

Шаг 1. Из вершины V_S в вершину V_T пересылается как можно больше единиц потока, полная стоимость прохождения по графу каждой из которых равна нулю.

Шаг 2. На следующем шаге из вершины V_S в вершину V_T пересылается как можно больше единиц потока, полная стоимость прохождения по графу каждой из которых равна единице и так далее.

Полная стоимость каждой единицы потока равна разности суммы стоимостей прямых дуг и суммы стоимостей обратных дуг.

Шаг 3. В алгоритме поиска потока минимальной стоимости используется в качестве подалгоритма «Алгоритм поиска увеличивающей цепи», где цепь увеличивается на нужное количество единиц потока. В конце алгоритма можно подсчитать минимальную общую стоимость для k единиц потока:

$$P(k) = \min \left\{ \sum_{\langle V_i, V_j \rangle \in E} a(V_i, V_j) \times f(V_i, V_j) \right\}.$$

Конец алгоритма.

¹³ \mathbf{Z} - множество целых чисел.

Пример 49. Для графа $G = \langle V, E \rangle$ (Рис. 78) передать k единиц потока с минимальной стоимостью. Рассмотреть случаи $k = \overline{1;4}$. Найти минимальную стоимость для каждого случая. Начальная вершина – $V_s = V_1$ и конечная вершина – $V_t = V_4$.

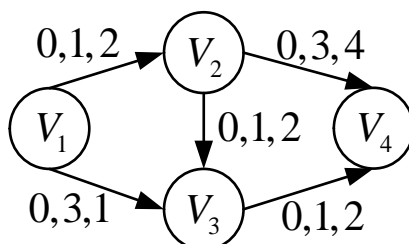


Рис. 78. Исходный граф.

1. Так как не может быть передана ни одна единица потока с нулевой стоимостью, а также со стоимостью 1 и 2, то, по цепи $E_1 = \{ \langle V_1, V_2 \rangle, \langle V_2, V_3 \rangle, \langle V_3, V_4 \rangle \}$ можно передать с минимальной стоимостью 2 единицы, каждая из которых будет иметь стоимость $P_1 = P_2 = 1 + 1 + 1 = 3$, где P_1 – стоимость первой единицы. Тогда $P(1) = 3$. После передачи первой единицы потока по графу получаем граф, изображенный на Рис. 79.

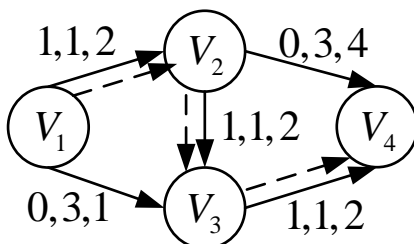


Рис. 79. Граф с одной единицей потока.

2. В пункте 1 вторая единица потока имеет такой же суммарный вес, как и первая. По пути

$E_2 = \{ \langle V_1, V_2 \rangle, \langle V_2, V_3 \rangle, \langle V_3, V_4 \rangle \}$ передаем вторую единицу потока $P_2 = 1+1+1=3$. В случае $k=2$ общая стоимость будет составлять $P(2) = P_1 + P_2 = 3+3=6$ – общая стоимость за две единицы потока. Результирующий граф представлен на Рис. 80.

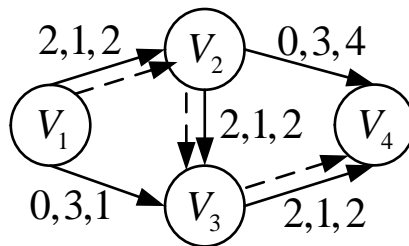


Рис. 80. Граф с двумя единицами потока.

3. Для случая $k=3$, можно использовать все предыдущие вычисления и добавить еще одну единицу потока с минимальной стоимостью. Из вершины $V_s = V_1$ уже нельзя перейти в вершину V_2 , следовательно, путь проходит через вершину V_3 . Из вершины V_3 нельзя попасть сразу в вершину $V_T = V_4$, значит необходимо переходить по обратной дуге $\langle V_2, V_3 \rangle$, уменьшая её поток на единицу, и из вершины V_2 в вершину $V_T = V_4$. Увеличивающая цепь для третьей единицы $E_3 = \{ \langle V_1, V_3 \rangle, \langle V_2, V_3 \rangle, \langle V_2, V_4 \rangle \}$ (Рис. 81).

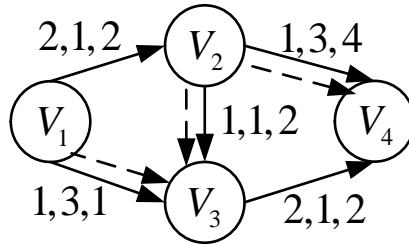


Рис. 81. Граф с тремя единицами потока минимальной стоимости.

$P_3 = 3 - 1 + 3 = 5$, $P(3) = P_1 + P_2 + P_3 = 6 + 5 = 11$ — общая стоимость трех единиц потока минимальной стоимости.

С другой стороны $P(k) = \min \{a(V_i, V_j) \times f(V_i, V_j)\}$,

$$P(3) = 2 \times 1 + 1 \times 3 + 1 \times 1 + 1 \times 3 + 2 \times 1 = 2 + 3 + 1 + 3 + 2 = 11.$$

4. При передаче трех единиц потока дуги графа, выходящие из источника, насыщаются. Дуга $\langle V_1, V_2 \rangle$ может передать только 2 единицы потока, а дуга $\langle V_1, V_3 \rangle$ только 1 единицу, то есть $K_{\max} = \sum_{V_j \in V} c(V_1, V_j) = 3$, $3 < 4 \Rightarrow 4$ единицы потока передать невозможно.

Ответ. $P(1) = 3$, $P(2) = 6$, $P(3) = 11$;

нельзя передать 4 единицы потока.

Тема 13. Задача почтальона для орграфов. Алгоритм поиска оптимального маршрута почтальона для орграфов

Задача почтальона для орграфов

Аналогами задачи почтальона в повседневной жизни являются задачи:

- ✓ обход железнодорожных путей,
- ✓ патрулирование улиц,
- ✓ совершение покупок,
- ✓ доставка товара,
- ✓ задачи маршрутизации.

Задача почтальона в терминах теории графов формулируется следующим образом: «В мультиграфе проверить возможность существования ормаршрута из произвольной вершины V_i , включающего все ребра графа ровно один раз и заканчивающегося также вершиной V_i ». Такой маршрут называется эйлеровым и знаком читателю из прошлых лекций. Соответствующий маршрут в орграфе принято называть маршрутом почтальона.

Условия существования маршрута почтальона

- ✓ Для неорграфа. Степень каждой вершины должна быть четной $\delta(V_i) = 2k, k \in \mathbf{N}, \forall V_i \in \mathbf{V}$.
- ✓ Для орграфа. Отрицательная и положительная степени вершин должны быть равны. $\delta^+(V_i) = \delta^-(V_i), \forall V_i \in \mathbf{V}$.

Определение. Орграф, в котором для каждой вершины ее отрицательная степень равна положительной степени, называется симметричным орграфом.

Определение. Смешанным графом называется граф, в котором встречаются и неориентированные ребра, и ориентированные дуги.

В данном пособии читателю предлагается изучать алгоритм

поиска оптимального маршрута почтальона для ориентированных графов.

Алгоритм поиска оптимального маршрута почтальона для орграфов

Начало. Дан граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – взвешенный мультиграф, в котором задана вершина V_0 , являющаяся началом обхода маршрута почтальона.

Шаг 1. Если $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – несимметричный граф, то перейти к шагу 2, иначе к шагу 3.

Шаг 2. Путем дублирования дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$ уравнять степени вершин, то есть сделать граф симметричным, причем суммарная длина дублируемых дуг должна быть минимальной.

Шаг 3. Построить эйлеров маршрут из заданной вершины V_0 , используя алгоритм поиска эйлерова цикла с выбором вершины по нумерации без повторных обходов. При возврате в заданную вершину V_0 , обойдя все дуги, получается ормаршрут почтальона для графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$.

Конец. Ормаршрут почтальона найден.

Более детально опишем шаг 2

1. $\forall V_i \in \mathbf{V} \exists \delta^+(V_i)$ (количество выходящих из вершины дуг) и $\exists \delta^-(V_i)$ (количество входящих в вершину дуг). Пусть $D(V_i) = \delta^-(V_i) - \delta^+(V_i)$. Согласно полученным значениям вершины делятся на три категории.
 - а. Если значение $D(V_i) = 0$, то вершина V_i – промежуточная.
 - б. Если значение $D(V_i) > 0$, то есть $\delta^-(V_i) > \delta^+(V_i)$, то вершина V_i – источник.
 - с. Если значение $D(V_i) < 0$, то есть $\delta^-(V_i) < \delta^+(V_i)$, то

вершина V_i – сток.

2. В в граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$ вводится дополнительный источник V_S и дополнительный сток V_T .

а. Дополнительный источник V_S соединяется дугами со всеми источниками, найденными в пункте 1б (Рис. 82).

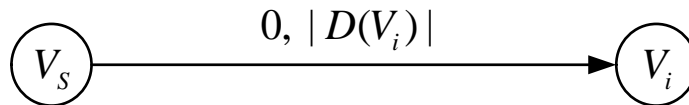


Рис. 82. Иллюстрация к пункту 2а.

Значение пропускной способности (максимальный поток) каждой дуги, выходящей из дополнительного источника V_S равно $|D(V_i)|$. Стоимость прохождения единицы потока равна 0.

б. Все вершины-стоки, найденные в пункте 1с соединяем с дополнительным стоком V_T . Значение пропускной способности каждой дуги, входящей в дополнительный сток V_T равно $|D(V_i)|$. Стоимость прохождения единицы потока по дуге равна 0 (Рис. 83).



Рис. 83. Иллюстрация к пункту 2б.

3. Для всех дуг графа $G = \langle \mathbf{V}, \mathbf{E} \rangle$ пропускная способность дуг $c \langle V_i, V_j \rangle := \infty$, а стоимость прохождения по дугам единицы потока прежняя.

4. Применяя к полученному графу алгоритм поиска максимального потока минимальной стоимости из источника V_S в сток V_T , можно получить новые дублируемые дуги, которые сделают граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$ симметричным.

5. Удаляются дополнительный источник V_S и дополнительный

- сток V_T , и дуги, инцидентные им.
 6. Граф $G = \langle \mathbf{V}, \mathbf{E} \rangle$ – симметричен.

Пример 50. Построить оптимальный маршрут почтальона, начиная с вершины V_1 для графа G , представленного на Рис. 84.

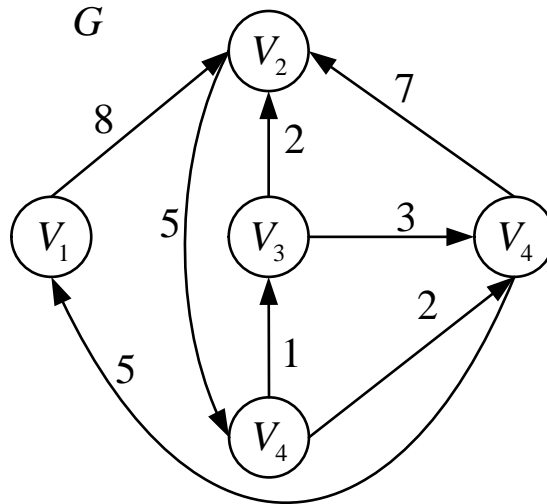


Рис. 84. Исходный граф.

Цифра на ребре обозначает стоимость прохождения по дуге одной единицы потока.

Решение приведено ниже по шагам согласно алгоритму поиска оптимального маршрута почтальона для орграфа.

1. Строим Табл. 8.

Табл. 8. Описание категорий вершин исходного графа.

$\delta^-(V_i)$	$\delta^+(V_i)$	Описание
1	1	Промежуточная
3	1	Источник
1	2	Сток
1	2	Сток
2	2	Промежуточная

2. Вводятся дополнительные источник V_S и сток V_T (Рис. 85).

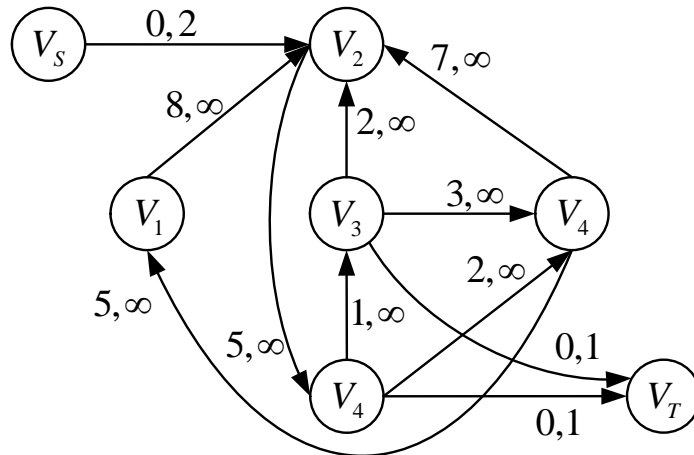


Рис. 85. Дополнительные источник и сток для графа.

Соединяем дополнительный источник с вершинами-источниками, а вершины-стоки соединяем с дополнительным стоком (Рис. 86).

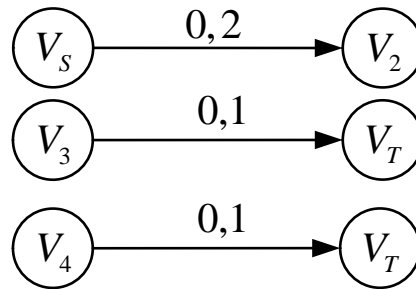


Рис. 86. Заполнение значений на новых дугах.

3. Из дополнительного источника проводится поток минимальной стоимости до насыщения новых дуг.
- $V_S \rightarrow V_2 \rightarrow V_4 \rightarrow V_T$ (дуга $\langle V_4, V_T \rangle$ насыщена).

b. $V_S \rightarrow V_2 \rightarrow V_4 \rightarrow V_3 \rightarrow V_T$ (дуги $\langle V_S, V_2 \rangle$ и $\langle V_3, V_T \rangle$ также насыщены).

В процессе насыщения дуг можно наблюдать последовательный проход по вершинам через дуги исходного графа. Эти дуги будут дублироваться.

Достраиваются три дополнительные дуги $\langle V_2, V_4 \rangle$, $\langle V_2, V_4 \rangle$, $\langle V_4, V_3 \rangle$.

4. Для получения симметричного графа, исключаются дуги $\langle V_S, V_2 \rangle$, $\langle V_3, V_T \rangle$, $\langle V_4, V_T \rangle$ (Рис. 87).

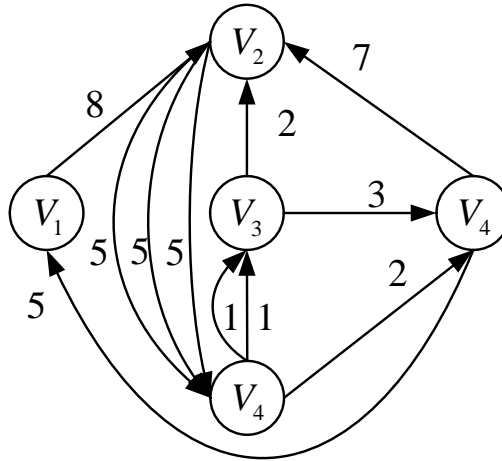


Рис. 87. Результирующий граф.

5. Начиная с вершины V_1 , строится эйлеров маршрут, вершины выбираются по нумерации. Читателю рекомендуется самостоятельно проделать эту несложную процедуру.

Ответ. $V_1 V_2 V_4 V_3 V_2 V_4 V_3 V_5 V_2 V_4 V_5 V_1$.

II. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

1. Словарь (глоссарий) основных терминов и понятий

Вершина графа – Узел, соединяющий рёбра и/или дуги. Множество вершин графа G обозначается $V(G)$.

Вес ребра — значение, поставленное в соответствие данному ребру взвешенного графа.

Взвешенный граф — граф, каждому ребру которого поставлено в соответствие некое значение (вес ребра).

Гамильтонов граф — граф, в котором есть гамильтонов цикл.

Гамильтонов путь — простой путь в графе, содержащий все вершины графа ровно по одному разу.

Гамильтонов цикл — простой цикл в графе, содержащий все вершины графа ровно по одному разу.

Граф — базовое понятие. Включает множество вершин и множество рёбер, являющееся подмножеством декартова множества вершин (то есть каждое ребро соединяет ровно две вершины).

Дерево — связный граф, не содержащий циклов.

Диаметр графа — это максимум расстояния между вершинами для всех пар вершин. Расстояние между вершинами — наименьшее число рёбер пути, соединяющего две вершины.

Длина маршрута — количество рёбер в маршруте (с повторениями)

Длина пути — число дуг пути (или сумма длин его дуг, если последние заданы). Так для пути v_1, v_2, \dots, v_n длина равна $n-1$.

Дуга — это ориентированное ребро.

Изолированная вершина — вершина, степень которой равна 0 (то есть нет ребер инцидентных ей).

Изоморфизм. Два графа называются **изоморфными**, если существует взаимно-однозначное соответствие между их вершинами и рёбрами, которое сохраняет смежность и инцидентность (графы отличаются только названиями своих вершин).

Инцидентность — понятие, используемое только в отношении ребра и вершины: если v_1, v_2 — вершины, а $e = (v_1, v_2)$ — соединяющее их ребро, тогда вершина v_1 и ребро e инцидентны, вершина v_2 и ребро e тоже инцидентны. Две вершины (или два ребра) инцидентными быть не могут. Для обозначения ближайших вершин (рёбер) используется понятие смежности.

Компонента связности графа — некоторое подмножество вершин графа такое, что для любых двух вершин из этого множества существует путь из одной в другую, и не существует пути из вершины этого множества в вершину не из этого множества.

Контур — замкнутый путь в орграфе.

Кратные рёбра — несколько рёбер, инцидентных одной и той же паре вершин. Встречаются в мультиграфах.

Кубический граф — регулярный граф степени 3, то есть граф в котором каждой вершине инцидентно ровно три ребра.

k-связный граф — связный граф, в котором не существует набора V' из $k - 1$ или менее вершин, такого, что удаление всех вершин V' и инцидентных им рёбер нарушает связность графа.

Лес — неориентированный граф без циклов. Компонентами связности леса являются деревья.

Локальная степень вершины — число рёбер ей инцидентных. Петля даёт вклад, равный «2» в степень вершины.

Матрица достижимости орграфа — это матрица, содержащая информацию о существовании путей между вершинами в орграфе.

Матрица инцидентности графа — это матрица, значения элементов которой характеризуется инцидентностью

соответствующих вершин графа (по вертикали) и его рёбер (по горизонтали). Для неориентированного графа элемент принимает значение **1**, если соответствующие ему вершина и ребро инцидентны. Для ориентированного графа элемент принимает значение **1**, если инцидентная вершина является началом ребра, значение **-1**, если инцидентная вершина является концом ребра; в случае петли **2**; в остальных случаях значению элемента присваивается **0**.

Матрица смежности графа — это матрица, значения элементов которой характеризуются смежностью вершин графа. При этом значению элемента матрицы присваивается количество рёбер, которые соединяют соответствующие вершины (то есть которые инцидентны обоим вершинам). Петля считается сразу двумя соединениями для вершины, то есть к значению элемента матрицы в таком случае следует прибавлять **2**.

Орграф, ориентированный граф $G = (V, E)$ есть пара множеств, где V — множество вершин, E — множество дуг (ориентированных рёбер). Дуга — это упорядоченная пара вершин (v, w) , где вершину v называют началом, а w — концом дуги.

Петля — ребро, начало и конец которого находятся в одной и той же вершине.

Планарный граф — граф, который может быть изображён на плоскости без пересечения рёбер.

Подграф исходного графа — граф, содержащий некоторое подмножество вершин данного графа и некоторое подмножество инцидентных им рёбер.

Полным графом называется граф, в котором для каждой пары вершин v_1, v_2 , существует ребро, инцидентное v_1 и инцидентное v_2 (каждая вершина соединена ребром с любой другой вершиной)

Простая цепь — маршрут, в котором все вершины различны.

Простой граф — граф, в котором нет кратных рёбер и петель.

Простой путь — путь, все рёбра которого попарно различны.

Простой цикл — цикл, не проходящий дважды через одну вершину.

Псевдограф — граф, содержащий петли и/или кратные рёбра.

Путь — последовательность рёбер (в неориентированном графе) и/или дуг (в ориентированном графе), такая, что конец одной дуги (ребра) является началом другой дуги (ребра). Или последовательность вершин и дуг (рёбер) в которой каждый элемент инцидентен предыдущему и последующему.

Путь в орграфе — это последовательность вершин v_1, v_2, \dots, v_n , для которой существуют дуги $v_1 \rightarrow v_2, v_2 \rightarrow v_3, \dots, v_{n-1} \rightarrow v_n$.

Радиус графа — минимальный из эксцентриситетов вершин связного графа; вершина, на которой достигается этот минимум называется центральной вершиной.

Разрез — множество ребер, удаление которого делает граф несвязным.

Расстояние между вершинами — длина кратчайшей цепи (в орграфе пути), соединяющей заданные вершины. Если такой цепи (пути) не существует, расстояние полагается равным бесконечности.

Ребро — базовое понятие. Ребро соединяет две вершины графа.

Связный граф — граф, в котором все вершины связаны.

Смежность — понятие, используемое в отношении только двух рёбер либо только двух вершин: Два ребра, инцидентные одной вершине, называются **смежными**; две вершины, инцидентные одному ребру, также называются **смежными**.

Смешанный граф — граф, содержащий как ориентированные, так и неориентированные рёбра.

Центр графа — вершина, расстояние от которой до самой дальней вершины минимально.

Цепь в графе — маршрут, все рёбра которого различны. Если все вершины (а тем самым и рёбра) различны, то такая цепь называется **простой**. В цепи $v_0, e_1, \dots, e_k, v_k$ вершины v_0 и v_k

называются **концами** цепи. Цепь с концами u и v **соединяет** вершины u и v .

Цикл — замкнутая цепь. Для орграфов цикл называется контуром.

Эйлеров граф — это граф, в котором существует цикл, содержащий все рёбра графа по одному разу (вершины могут повторяться).

Эйлерова цепь (или **Эйлеров цикл**) — это цепь (цикл), которая содержит все рёбра графа (вершины могут повторяться).

Эксцентриситет вершины — максимальное из расстояний от данной вершины до других вершин.

2. Методические указания для преподавателя, студента, слушателя

Методические рекомендации по организации изучения дисциплины:

На освоение дисциплины отводится 1 семестр. В качестве итогового контроля знаний предусмотрен экзамен. Упражнения, лекции, методику, тестирования и задания для самостоятельной работы студентов можно увидеть в разделе «Фонды оценочных средств» данного учебно-методического комплекса.

3. Сборник задач и упражнений

Отсутствует

4. Лабораторный практикум по дисциплине

Тема: Основные понятия теории графов.

Неориентированные графы

Семинар 1.

№ 1. Дан граф $G(\mathbf{V}, \mathbf{E})$ (Рис. 88):

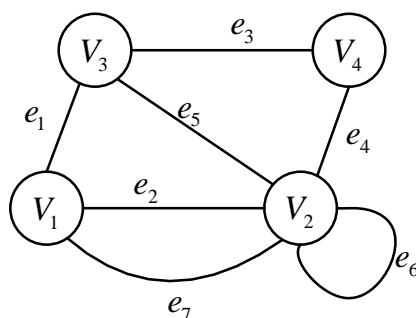


Рис. 88. Исходный граф.

Определить:

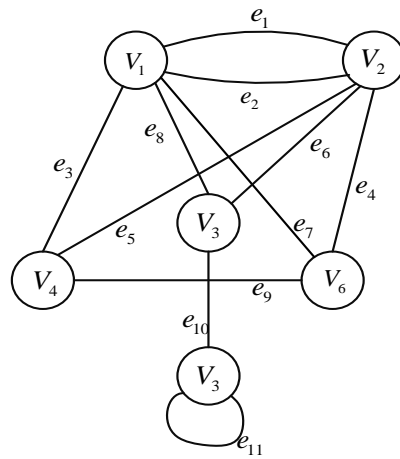
- 1) Множество вершин \mathbf{V} и множество ребер \mathbf{E} .
- 2) Пары смежных вершин.
- 3) Инцидентность ребра вершинам.
- 4) Пары смежных ребер.
- 5) Степени вершин.
- 6) Параллельные ребра.
- 7) Наличие петель.

№ 2. Построить неизоморфные графы с пятью вершинами и ребрами, $i = \overline{0,10}$.

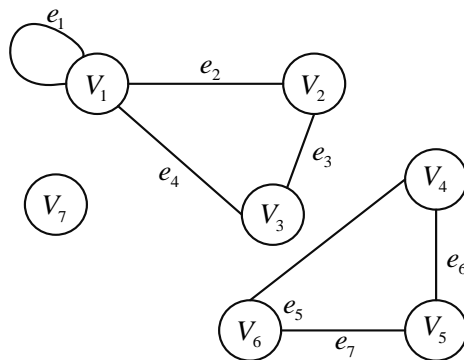
Семинар 2.

№ 1. Построить матрицы смежности и инцидентности для графов:

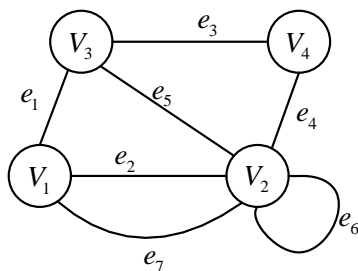
a)



б)

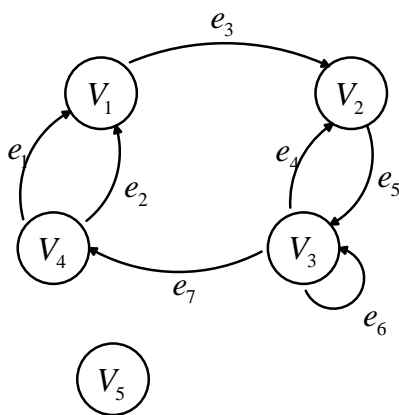


в) Для графа составить матрицу смежности \mathbf{B} и матрицу инцидентности \mathbf{A} .



Замечание: Если в графе отсутствуют петли и кратные ребра, то для доказательства изоморфизма достаточно сравнить их матрицы смежности или инцидентности.

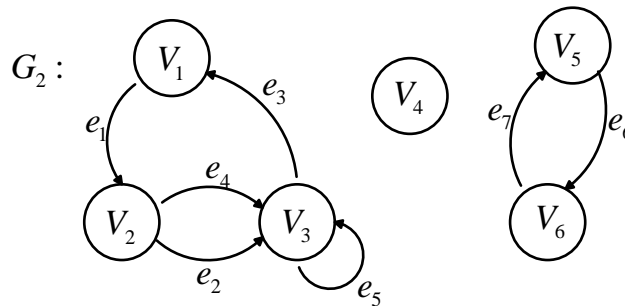
№ 2. Дан граф $G(\mathbf{V}, \mathbf{E})$.



Найти:

- 1) Множество вершин \mathbf{V} и множество дуг \mathbf{E} .
- 2) Пары смежных вершин.
- 3) Положительная инцидентность дуг вершинам.
- 4) Отрицательная инцидентность дуг вершинам.
- 5) Наличие петель.
- 6) Наличие строго параллельных дуг.
- 7) Наличие нестрого параллельных дуг.
- 8) Пары смежных дуг.
- 9) Положительные и отрицательные степени вершин. Степень вершин $\delta(V) = \delta_{(V)}^+ + \delta_{(V)}^-$.
- 10) Количество вершин нечетной степени.
- 11) A – матрицу инцидентности.
- 12) B – матрицу смежности.

№ 3. Дан граф $G_2(\mathbf{V}, \mathbf{E})$.



Найти:

- 1) Множество вершин \mathbf{V} и множество дуг \mathbf{E} .
- 2) Пары смежных вершин.

- 3) Положительная инцидентность дуг вершинам.
- 4) Отрицательная инцидентность дуг вершинам.
- 5) Наличие петель.
- 6) Наличие строго параллельных дуг.
- 7) Наличие нестрого параллельных дуг.
- 8) Пары смежных дуг.
- 9) Положительные и отрицательные степени вершин. Степень вершин $\delta(V) = \delta_{(V)}^+ + \delta_{(V)}^-$.
- 10) Количество вершин нечетной степени.
- 11) **A** – матрицу инцидентности.
- 12) **B** – матрицу смежности.

№ 4. Даны матрица инцидентности и матрица смежности для графа. Составить схему графа.

$$A_1 = \begin{pmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \\ \hline V_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline V_2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline V_3 & 0 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ \hline V_4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ \hline V_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline V_6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \\ \hline V_1 & 0 & 1 & 2 & 0 & 0 & 0 \\ \hline V_2 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline V_3 & 2 & 1 & 1 & 0 & 0 & 0 \\ \hline V_4 & 0 & 0 & 0 & 0 & 1 & 2 \\ \hline V_5 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline V_6 & 0 & 0 & 0 & 2 & 1 & 0 \end{pmatrix}$$

№ 5. Дана матрица инцидентности A_2 .

Построить:

1) схему графа.

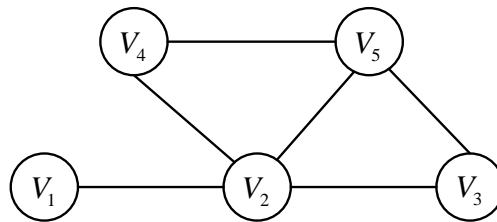
2) матрицу смежности B_2 .

$$A_2 = \begin{pmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \\ V_1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ V_2 & -1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ V_3 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ V_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 \\ V_5 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ V_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ V_7 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ V_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Тема: Метрические характеристики графов

Семинар 3.

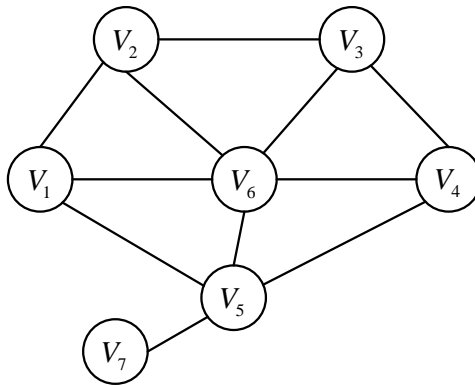
№ 1. Дан граф.



Найти:

- 1) Расстояние между различными вершинами (кратчайшая простая цепь, т.е. все вершины различны, все веса за 1 берем), $d(V_i, V_j) = ?$ ($i \neq j$).
- 2) Эксцентриситет каждой вершины $l(V_i) = \max_{V_j \in V} d(V_i, V_j)$.
- 3) Диаметр графа $d(G) = \max_{V \in V} l(V)$.
- 4) Радиус графа $r(G) = \min_{V \in V} l(V)$.
- 5) Определить центральные вершины графа $l(V_i) = r(G)$.
- 6) Определить центр графа.
- 7) Найти число маршрутов длины 2 для всех вершин графа (т.е. \mathbf{B}^2).
- 8) Найти число маршрутов длины 3 для всех вершин графа (т.е. \mathbf{B}^3).
- 9) Определить список смежности для графа G .

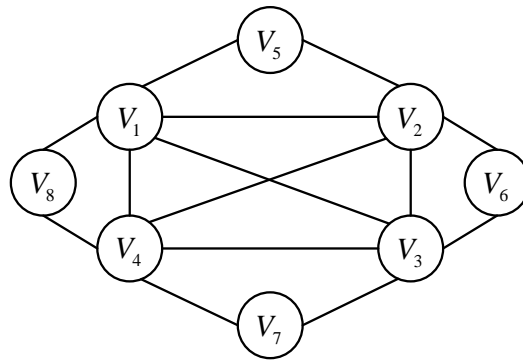
№ 2. Дан граф.



Найти:

- 1) Расстояние между различными вершинами (кратчайшая простая цепь, т.е. все вершины различны, все веса за 1 берем), $d(V_i, V_j) = ?$ ($i \neq j$).
- 2) Эксцентриситет каждой вершины $l(V_i) = \max_{V_j \in V} d(V_i, V_j)$.
- 3) Диаметр графа $d(G) = \max_{V \in V} l(V)$.
- 4) Радиус графа $r(G) = \min_{V \in V} l(V)$.
- 5) Определить центральные вершины графа $l(V_i) = r(G)$.
- 6) Определить центр графа.
- 7) Найти число маршрутов длины 2 для всех вершин графа (т.е. \mathbf{B}^2).
- 8) Найти число маршрутов длины 3 для всех вершин графа (т.е. \mathbf{B}^3).
- 9) Определить список смежности для графа G .

№ 3. Дан граф.



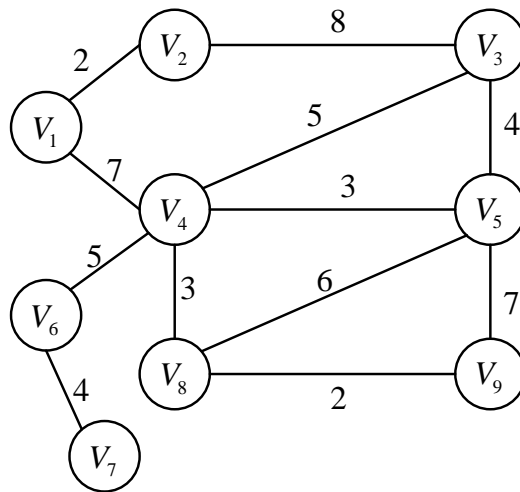
Найти:

- 1) Расстояние между различными вершинами (кратчайшая простая цепь, т.е. все вершины различны, все веса за 1 берем), $d(V_i, V_j) = ? \quad (i \neq j)$.
- 2) Эксцентриситет каждой вершины $l(V_i) = \max_{V_j \in V} d(V_i, V_j)$.
- 3) Диаметр графа $d(G) = \max_{V \in V} l(V)$.
- 4) Радиус графа $r(G) = \min_{V \in V} l(V)$.
- 5) Определить центральные вершины графа $l(V_i) = r(G)$.
- 6) Определить центр графа.
- 7) Найти число маршрутов длины 2 для всех вершин графа (т.е. \mathbf{B}^2).
- 8) Найти число маршрутов длины 3 для всех вершин графа (т.е. \mathbf{B}^3).
- 9) Определить список смежности для графа G .

Тема: Алгоритм Краскала

Семинар 4.

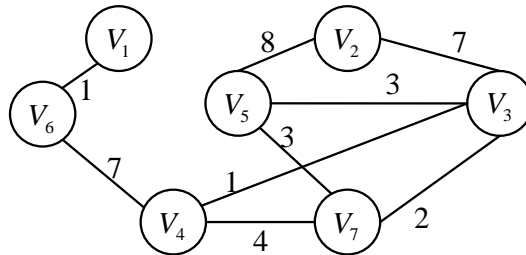
№ 1. Дан граф G .



Для графа построить по алгоритму Краскала:

- 1) минимальное покрывающее дерево и найти вес $w_{T_{\min}}$ полученного дерева;
- 2) максимальное покрывающее дерево и найти вес $w_{T_{\max}}$ полученного дерева.

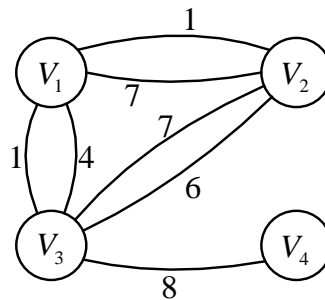
№ 2. Дан граф G .



Для графа построить по алгоритму Краскала:

- 1) покрывающее дерево по нумерации;
- 2) минимальное покрывающее дерево и найти вес $w_{T_{\min}}$ полученного дерева;
- 3) максимальное покрывающее дерево и найти вес $w_{T_{\max}}$ полученного дерева.

№ 3. Дан граф G .



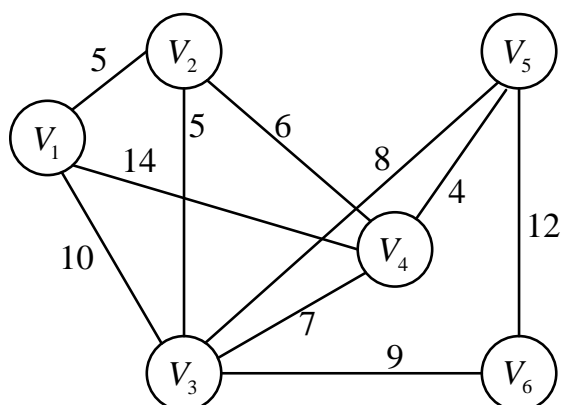
Для графа построить по алгоритму Краскала:

- 1) покрывающее дерево по нумерации;
- 2) минимальное покрывающее дерево и найти вес $w_{T_{\min}}$ полученного дерева;
- 3) максимальное покрывающее дерево и найти вес $w_{T_{\max}}$ полученного дерева.

Тема: Алгоритм Прима

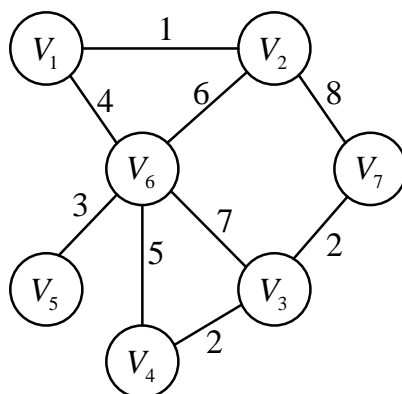
Семинар 5.

№ 1. Дан граф G .



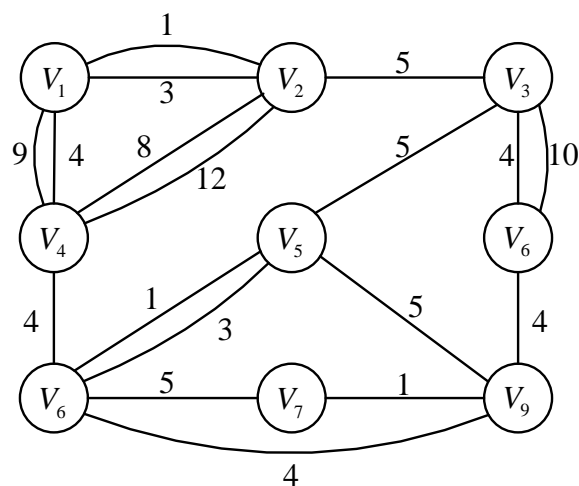
Построить минимальное покрывающее дерево по алгоритму Прима и посчитать вес дерева $w_{T_{\min}}$.

№ 2. Дан граф G .



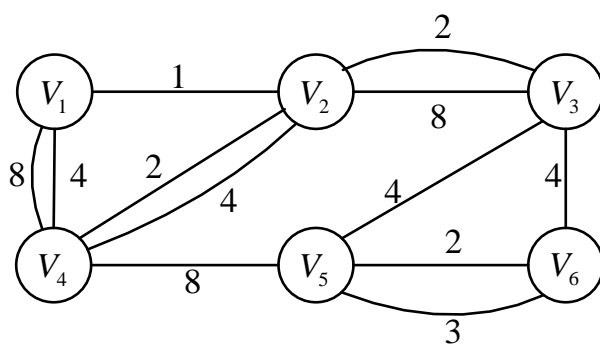
Построить максимальное покрывающее дерево по алгоритму Прима и посчитать вес дерева $w_{T_{\max}}$.

№ 3. Дан граф G .



Построить минимальное покрывающее дерево по алгоритму Прима и найти его вес $w_{T_{\min}}$.

№ 4. Дан граф G .

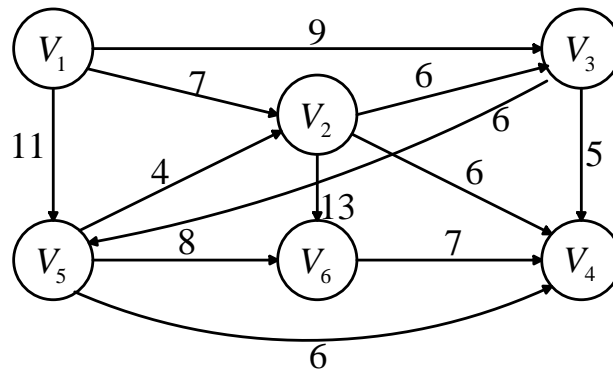


Построить максимальное и минимальное дерево по алгоритму Прима и для каждого посчитать веса $w_{T_{\max}}$ и $w_{T_{\min}}$. Изобразить результаты в виде деревьев с весами.

Тема: Алгоритм Дейкстры

Семинар 6.

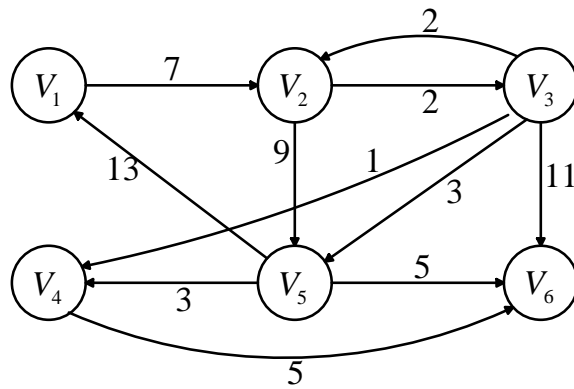
№ 1. Дан граф G .



Используя Алгоритм Дейкстры найти минимальный путь и длину:

- 1) от вершины V_1 до вершины V_4 ;
- 2) от вершины V_3 до вершины V_6 ;
- 3) от вершины V_1 до вершины V_6 .

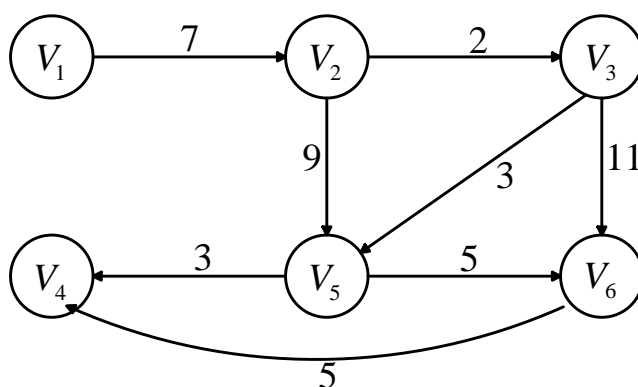
№ 2. Дан граф G .



Используя Алгоритм Дейкстры найти минимальный путь и длину:

- 1) от вершины V_2 до вершины V_4 ;
- 2) от вершины V_1 до вершины V_5 ;
- 3) от вершины V_1 до вершины V_4 .

№ 3. Дан граф G .



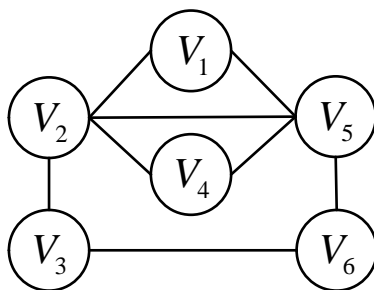
Используя Алгоритм Дейкстры найти минимальный путь и длину:

- 1) от вершины V_1 до вершины V_4 ;
- 2) от вершины V_1 до вершины V_6 ;
- 3) от вершины V_2 до вершины V_6 .

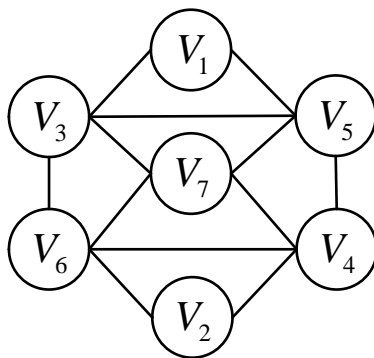
Тема: Поиск эйлера цикла

Семинар 7.

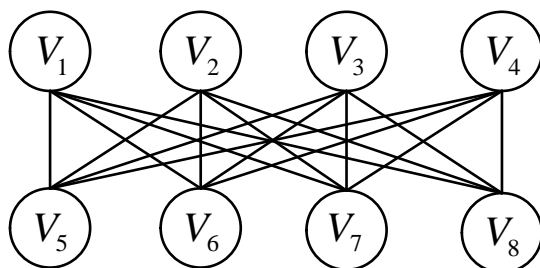
№ 1. Построить эйлеров цикл в графе, начиная с вершины V_4 и с вершины V_3 .



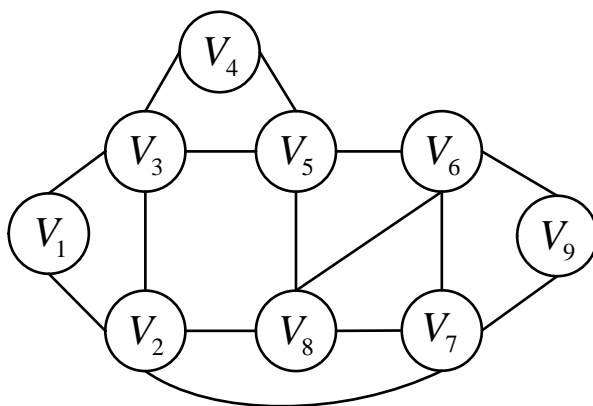
№ 2. Построить эйлеров цикл в графе, начиная с вершины V_1 .



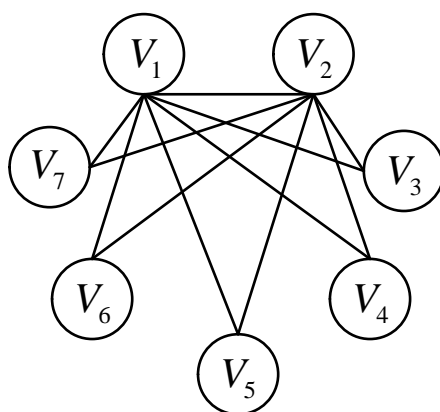
№ 3. Построить эйлеров цикл в графе, начиная с вершины V_8 .



№ 4. Построить эйлеров цикл в графе, начиная с вершины V_1 .



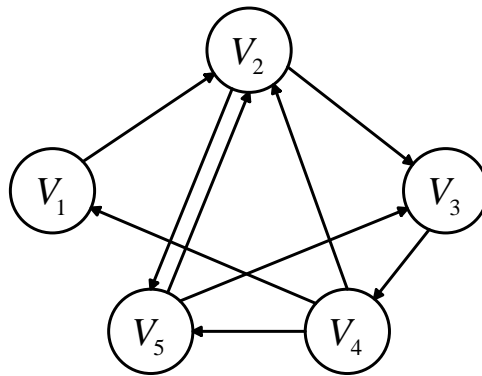
№ 5. Построить эйлеров цикл в графе, начиная с вершины V_4 .



Тема: Гамильтоновы графы

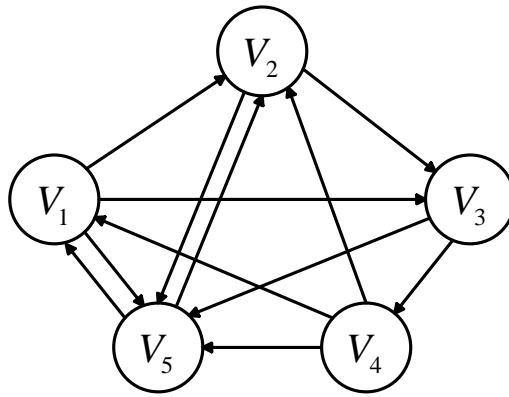
Семинар 8.

№ 1. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



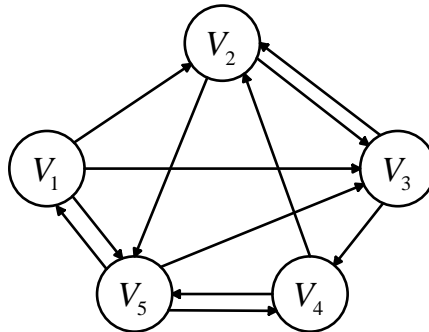
Из вершины V_2 найти всевозможные гамильтоновы циклы.

№ 2. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



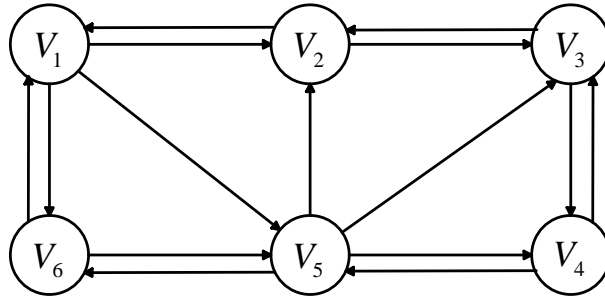
Найти количество гамильтоновых циклов из вершины V_2 и указать их.

№ 3. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



Найти количество гамильтоновых циклов из вершины V_3 и указать их.

№ 4. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.

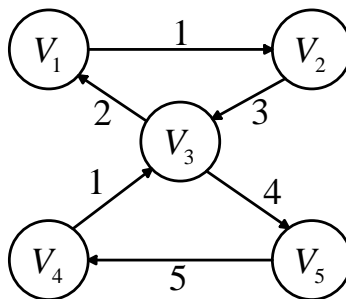


Из вершины V_1 найти всевозможные гамильтоновы циклы.

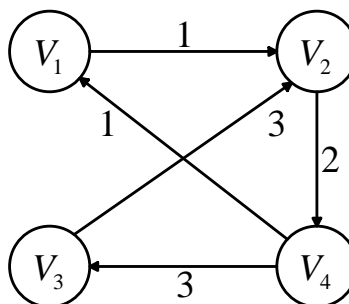
Тема: Алгоритм Уоршалла-Флойда. Задача транзитивного замыкания

Семинар 9.

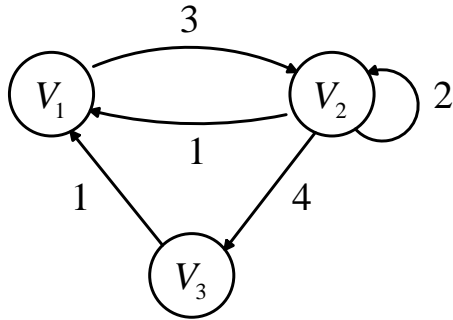
№ 1. Используя алгоритм Уоршалла-Флойда, найти минимальное расстояние между всеми парами вершин.



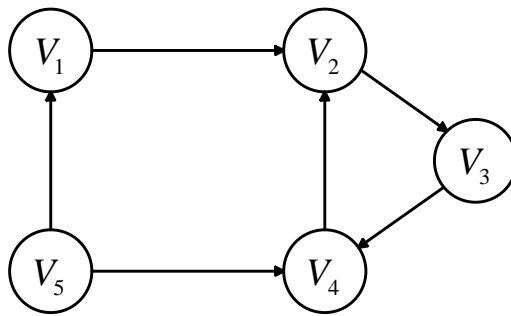
№ 2. Используя алгоритм Уоршалла-Флойда, найти минимальное расстояние между всеми парами вершин.



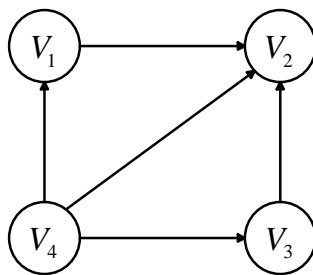
№ 3. Используя алгоритм Уоршалла-Флойда, найти минимальное расстояние между всеми парами вершин.



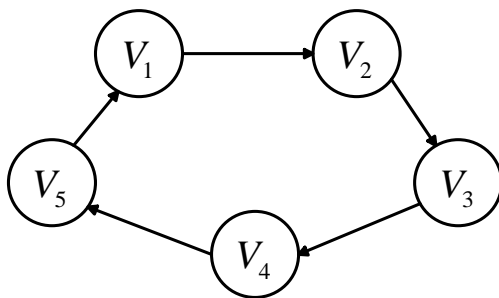
№ 4. Построить транзитивное замыкание для графа и найти матрицу связности графа.



№ 5. Построить транзитивное замыкание для графа и найти матрицу связности графа.



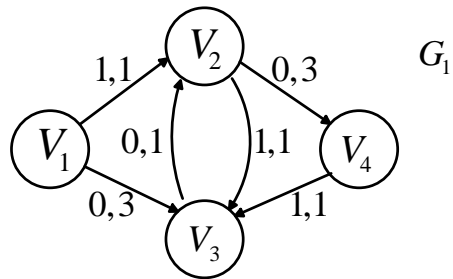
№ 6. Построить транзитивное замыкание для графа и найти матрицу связности графа.



Тема: Потоки

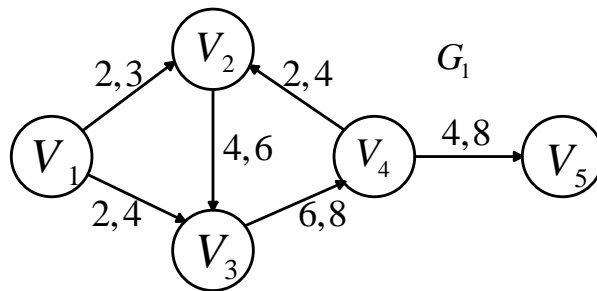
Семинар 10.

№ 1. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



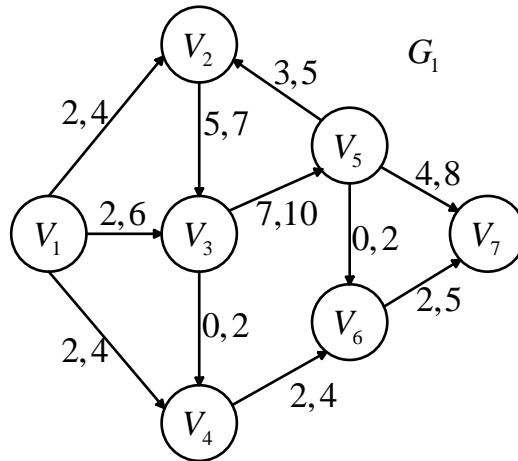
- 1) Проверить условие существования потока из вершины V_1 в вершину V_4 .
- 2) Найти увеличивающую цепь, если возможно, и увеличить поток.

№ 2. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



- 1) Проверить условие существования потока из вершины V_1 в вершину V_5 .
- 2) Найти увеличивающую цепь, если возможно, и увеличить поток.

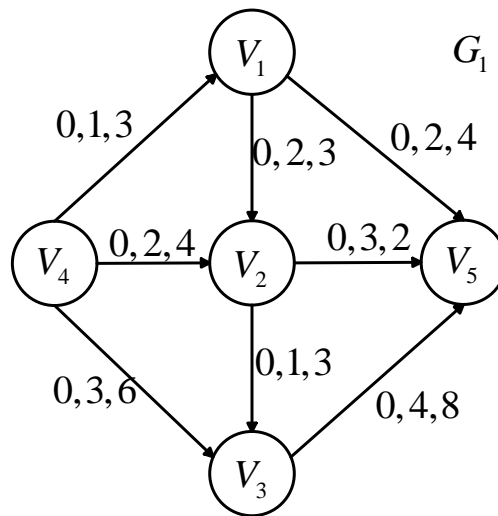
№ 3. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



- 1) Проверить условие существования потока из вершины V_1 в вершину V_7 .
- 2) Найти увеличивающую цепь, если возможно, и увеличить поток.

Семинар 11.

№ 1. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.

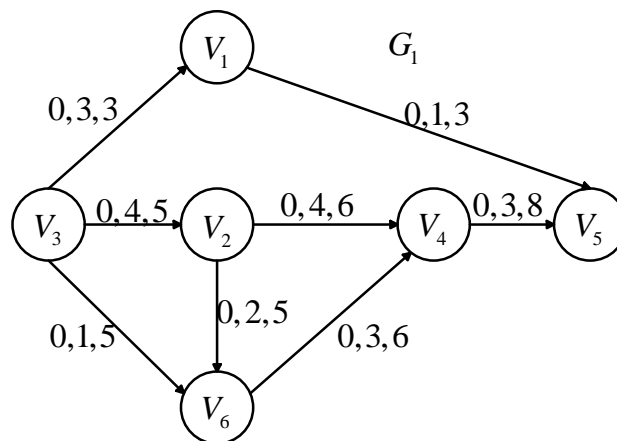


Найти:

- 1) максимальный поток K_{\max} на графе, где источник – вершина V_4 и сток – вершина V_5 ;
- 2) поток минимальной стоимости для $k = 2$, $k = 5$. Узнать стоимость передачи.

Изобразить итоговые графы.

№ 2. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.

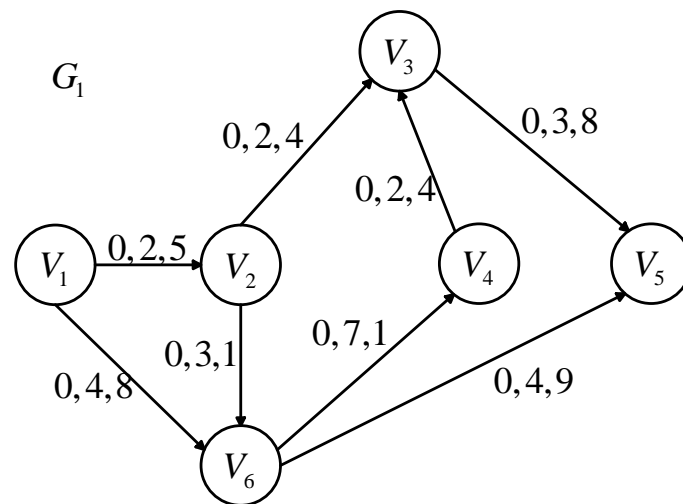


Найти:

- 1) максимальный поток K_{\max} на графе, где источник – вершина V_3 и сток – вершина V_5 ;
- 2) поток минимальной стоимости для $k = 2$, $k = 5$. Узнать стоимость передачи.

Изобразить итоговые графы.

№ 3. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



Найти:

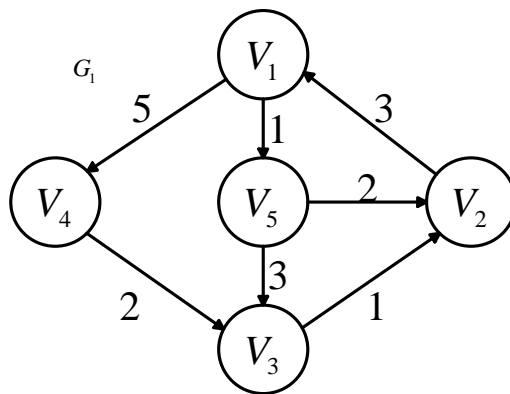
- 1) максимальный поток K_{\max} на графе, где источник – вершина V_1 и сток – вершина V_5 ;
- 2) поток минимальной стоимости для $k = 2$, $k = 5$. Узнать стоимости.

Изобразить итоговые графы.

Тема: Задача почтальона для орграфа

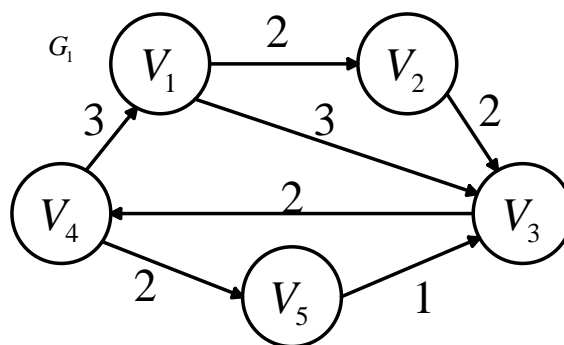
Семинар 12.

№ 1. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



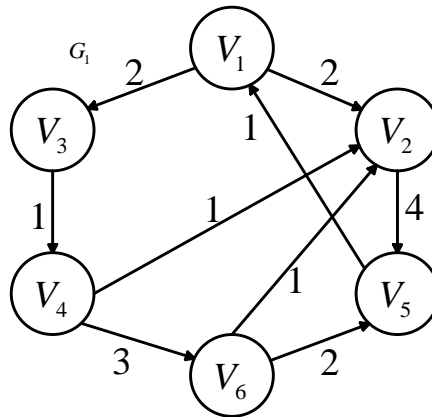
Построить оптимальный маршрут почтальона из вершины V_3 .

№ 2. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$



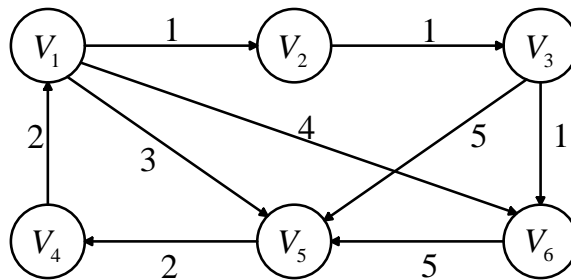
Построить оптимальный маршрут почтальона из вершины V_5 .

№ 3. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



Построить оптимальный маршрут почтальона из вершины V_2 .

№ 4. Дан граф $G_1 = \langle \mathbf{V}, \mathbf{E} \rangle$.



Построить оптимальный маршрут почтальона из вершины V_1 .

5. Описание балльно-рейтинговой системы

Работа в семестре

Максимальное число баллов, набранных в семестре – 100

Вид задания	Число заданий	Кол-во баллов	Сумма баллов
1. Посещение лекций	–	–	–
2. Лабораторные работы	–	–	–
3. Практические занятия	–	–	–
4. Домашние задания	–	–	–
5. Контрольные работы	2	40,40	80
6. Рубежная аттестация	–	–	–
7. Работа на семинаре	–	–	–
8. Реферат	–	–	–
9. Коллоквиум	–	–	–
10. Итоговая аттестация (экзамен)	1	20	20
ИТОГО			100

Соответствие систем оценок (используемых ранее оценок итоговой академической успеваемости, оценок ECTS и балльно-

рейтинговой системы (БРС) оценок текущей успеваемости) (В соответствии с Приказом Ректора №996 от 27.12.2006 г.):

Баллы БРС	Традиц. оценки в РФ	Баллы для перевода оценок	Оценки	Оценки ECTS
86 – 100	5	95 – 100	5+	A
		86 – 94	5	B
69 – 85	4	69 – 85	4	C
51 – 68	3	61 – 68	3+	D
		51 – 60	3	E
0 – 50	2	31 – 50	2+	FX
		0 – 30	2	F
51 – 100	Зачет		Зачет	Passed

Студенты обязаны сдавать все задания в сроки, установленные преподавателем. Работы, предоставленные с опозданием, не оцениваются. Студенты, получившие в течение семестра, оценку 3 или 4 (зачет) и желающие повысить свою оценку, допускаются к экзамену (итоговая аттестация). Экзаменационная работа оценивается из 20 баллов независимо от оценки, полученной в семестре. Оценка менее 51 балла (<3),

полученная при итоговой аттестации, является неудовлетворительной.

Студенты, набравшие менее 31 балла в течение семестра не допускаются к итоговой аттестации.

6. Вопросы для самопроверки и обсуждений по темам

Для текущего контроля успеваемости и промежуточной аттестации студентов рекомендуется использовать вопросы и задания подобные перечисленным ниже:

Типовые задачи для промежуточного контроля знаний:

1. Найти эксцентриситет, диаметр и радиус графа.
2. Составить матрицу смежности и матрицу инцидентности для графа.
3. Построить минимальное (максимальное) покрывающее дерево для графа, используя алгоритм Краскала.
4. Построить минимальное (максимальное) покрывающее дерево для графа, используя алгоритм Прима.
5. Найти Эйлеров цикл в графе.
6. Найти кратчайший путь и длину между заданными вершинами, используя алгоритм Дейкстры.
7. Найти минимальное расстояние между всеми парами вершин по алгоритму Уоршалла-Флойда.
8. Построить транзитивное замыкание и найти матрицу связности (достижимости) для графа. Представить решение через промежуточные матрицы.
9. Найти гамильтоновы циклы в графе из заданной вершины, используя алгоритм поиска гамильтонова цикла в графе.
10. Найти поток минимальной стоимости, состоящий из k (например, 5) единиц. (При решении обязательно указывать промежуточные графы и увеличивающие цепи. В ответе изобразить результирующий граф и указать минимальную стоимость потока.)
11. Построить оптимальный маршрут почтальона из заданной

вершины. (В решении построить симметричный граф, указать потоки минимальной стоимости. В ответе указать результирующий граф и оптимальный маршрут почтальона.)

12. Найти максимальный поток на графе. (Увеличивать поток с указанием увеличивающей цепи и количеством единиц, передаваемых по увеличивающей цепи. На каждом шаге изображать новый граф с последующей нумерацией графа.) В ответе указать максимальный поток и результирующий граф.

7.Задания для самостоятельной работы по темам

Для самостоятельной работы и подготовки к первой контрольной работы рекомендуется выполнить следующее задание:

1. Изобразить связный невзвешенный неорграф, состоящий из 5 вершин и более 6 ребер. Для этого неорграфа найти:
ЧАСТЬ 1
 - а) множества V и E ,
 - б) пары смежных вершин,
 - в) инцидентность ребра вершинам,
 - г) пары смежных ребер,
 - д) указать степени всех вершин,
 - е) указать параллельные ребра (при наличии),
 - ж) указать петли (при наличии),
 - з) составить для неорграфа матрицу смежности и список смежности,
 - и) составить для неорграфа матрицу инцидентности,
 - к) указать количество вершин нечетной степени.

Далее, удалив из графа все петли и кратные ребра (оставив одно ребро), получить граф, изобразить его на отдельном рисунке и найти для него: **ЧАСТЬ 2:**

- л) расстояния между всеми парами вершин,
 - м) эксцентриситет каждой вершины,
 - н) радиус графа,
 - о) диаметр графа,
 - п) определить центр графа,
 - р) найти число маршрутов длины 2 для графа.
2. Изобразить невзвешенный орграф, состоящий из 5 вершин и более 7 дуг. Для этого орграфа найти:
- а) множества V и E ,
 - б) пары смежных вершин,
 - в) положительную и отрицательную инцидентность дуг вершинам,
 - г) пары смежных дуг,
 - д) указать степени всех вершин, а также отрицательную и положительную степени,
 - е) указать строгопараллельные ребра и нестрогопараллельные ребра (при наличии),
 - ж) указать петли (при наличии),
 - з) составить для орграфа матрицу смежности,
 - и) составить для орграфа матрицу инцидентности,
 - к) указать количество вершин нечетной степени.
3. Изобразить связный взвешенный неорграф, состоящий из 6 вершин и более 10 ребер. Для этого задания выбрать такой граф, для которого степени каждой вершины – четные. Для графа найти:
- а) покрывающее дерево по алгоритму Краскала по нумерации,
 - б) минимальное покрывающее дерево по алгоритму Краскала,
 - в) максимальное покрывающее дерево по алгоритму Краскала,
 - г) минимальное покрывающее дерево по алгоритму Прима,
 - д) максимальное покрывающее дерево по алгоритму Прима,
 - е) построить эйлеров цикл в графе, начиная с 5 вершины.
4. Изобразить связный взвешенный орграф, состоящий из 7 вершин и более 10 дуг. Определить начальную и конечную вершины для поиска минимального расстояния и пути по

алгоритму Дейкстры, причем путей из начальной вершины в конечную должно быть на графе не менее двух. Определить из них минимальную длину и путь по алгоритму, введенному на лекции.

Для самостоятельной работы и подготовки к второй контрольной работы рекомендуется выполнить следующее задание:

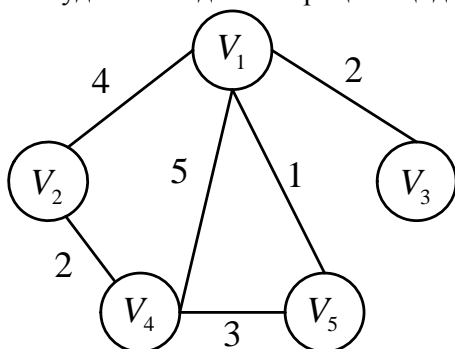
1. Построить орграф, содержащий не менее 6 вершин и 9 дуг. Найти транзитивное замыкание и матрицу связности для орграфа.
2. Построить взвешенный орграф, содержащий не менее 6 вершин и 9 дуг. По алгоритму Уоршалла-Флойда найти минимальное расстояние между всеми парами вершин.
3. Построить орграф, содержащий не менее 7 вершин и 9 дуг. В графе должно быть не менее 4 увеличивающих цепей, содержащих прямые и обратные дуги. Найти максимальный поток на графе, описывая условия существования потока на графе, вводя дополнительные графы.
4. Построить орграф, содержащий не менее 7 вершин и 9 дуг. В графе должно быть не менее 4 увеличивающих цепей, содержащих прямые и обратные дуги. Найти поток минимальной стоимости для 5 единиц. Поток должен проходить по нескольким цепям. Нужно описывать условия существования потока на графе, вводя дополнительные графы.
5. Построить орграф, содержащий не менее 7 вершин и 9 дуг. Построить оптимальный маршрут почтальона из вершины V_2 . Граф должен быть первоначально несимметричным. Применить для орграфа алгоритм поиска эйлера цикла по нумерации и найти единственно возможный вариант маршрута почтальона.
6. Построить орграф, содержащий не менее 7 вершин и 9 дуг. Найти гамильтоновы циклы в орграфе, начиная с вершины a . Циклов должно быть не менее 3.

8. Перечень рефератов и/или курсовых работ по темам

1. Построение минимального покрывающего дерева по алгоритму Прима.
2. Построение максимального покрывающего дерева по алгоритму Прима.
3. Построение минимального покрывающего дерева по алгоритму Краскала.
4. Построение максимального покрывающего дерева по алгоритму Краскала.
5. Построение Эйлера цикла в графе.
6. Построение Гамильтонова цикла в графе.
7. Поиск максимального потока в графе.
8. Поиск потока минимальной стоимости в графе.
9. Построение кратчайшего пути и поиск длины между заданными вершинами графа с использованием алгоритма Дейкстры.
10. Поиск минимального расстояния между всеми парами вершин по алгоритму Уоршалла-Флойда.
11. Построение транзитивного замыкания для графа.
12. Построение оптимального маршрута почтальона в графе.

9. Тестовые задания по темам (для текущего и промежуточного самоконтроля)

1. Как будет выглядеть матрица инцидентности для графа:



Ответ:

□

	(V_1, V_2)	(V_1, V_3)	(V_1, V_4)	(V_1, V_5)	(V_2, V_4)	(V_4, V_5)
V_1	1	1	1	1	0	0
V_2	1	0	0	0	1	0
V_3	0	1	0	0	0	0
V_4	0	0	1	0	1	1
V_5	0	0	0	1	0	1

□

	(V_1, V_2)	(V_1, V_3)	(V_1, V_4)	(V_1, V_5)	(V_2, V_4)	(V_4, V_5)
V_1	4	2	5	1	0	0
V_2	4	0	0	0	2	0
V_3	0	2	0	0	0	0
V_4	0	0	5	0	2	3
V_5	0	0	0	1	0	3

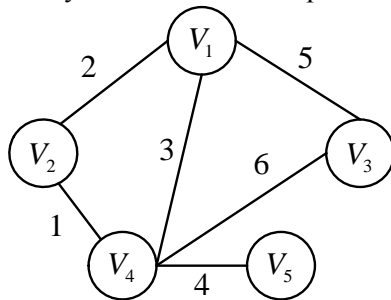
□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	0	1	1	1	1
V ₂	-1	0	0	1	0
V ₃	-1	0	0	0	0
V ₄	-1	-1	0	0	1
V ₅	-1	0	0	-1	0

□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	0	1	1	1	1
V ₂	1	0	0	1	0
V ₃	1	0	0	0	0
V ₄	1	1	0	0	1
V ₅	1	0	0	1	0

2. Как будет выглядеть матрица весов для графа:



Ответ:

□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	0	2	5	3	0
V ₂	2	0	0	1	0
V ₃	5	0	0	6	0
V ₄	3	1	6	0	4
V ₅	0	0	0	4	0

□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	0	2	5	3	∞
V ₂	2	0	∞	1	∞
V ₃	5	∞	0	6	∞
V ₄	3	1	6	0	4
V ₅	∞	∞	∞	4	0

□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	0	2	5	3	∞
V ₂	-2	0	∞	1	∞
V ₃	-5	∞	0	6	∞

V ₄	-3	-1	-6	0	4
V ₅	∞	∞	∞	-4	0

□

	V ₁	V ₂	V ₃	V ₄	V ₅
V ₁	1	-2	-5	-3	∞
V ₂	2	1	∞	-1	∞
V ₃	5	∞	1	-6	∞
V ₄	3	1	6	1	-4
V ₅	∞	∞	∞	4	1

3. Маршрут в графе можно задать:

Ответ: только последовательностью ребер;

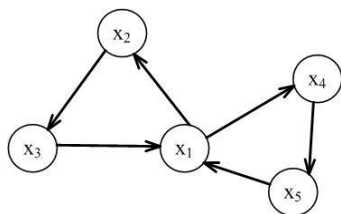
только последовательностью вершин;

и последовательностью ребер и последовательностью вершин;

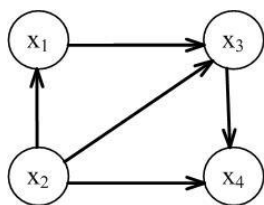
ни последовательностью ребер, ни последовательностью вершин.

4. Какие из графов являются связными?

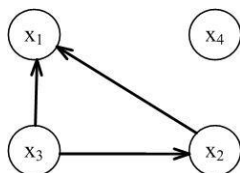
А



В



С



Ответ: А и В;

С;

А;

В.

5. Выберите правильные утверждения. Неориентированное дерево есть:

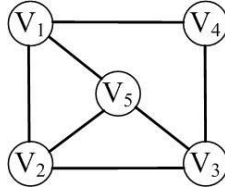
Ответ: связный граф, содержащий n вершин и $n-1$ ребер;

связный граф, содержащий n вершин и $n-1$ ребер, и не имеющий циклов;

граф, в котором любые две вершины соединены двумя цепями;

любое подмножество $n-1$ неориентированных ребер и n вершин.

6. Для графа, представленного на рисунке, число маршрутов длины 2 для всех вершин равно:



Ответ: 14;

0	1	0	1	1
1	0	1	0	1
0	1	0	1	1
1	0	0	0	0
1	1	1	0	0

3	1	3	0	1
1	3	1	2	2
3	1	3	0	1
0	2	0	2	2
1	2	1	2	3

2	7	2	6	7
7	4	7	2	5
2	7	2	6	7
6	2	6	0	2
7	5	7	2	4

7. Какие из утверждений справедливы?

Ответ: В конечном графе число вершин нечетной степени нечетно.

В неориентированном графе число вершин четной степени четно.

В конечном графе число вершин нечетной степени четно.

В орграфе число вершин четной степени четно.

8. Алгоритм Краскала может быть применен для :

Ответ: построения цикла;

поиска минимального дерева;

поиска кратчайшего пути между вершинами;

поиска радиуса графа.

9. При построении минимального покрывающего дерева по алгоритму Краскала используются:

Ответ: букет;

матрица инцидентности;

матрица смежности;

матрица весов.

10. При построении минимального покрывающего дерева по алгоритму Прима используются:

Ответ: матрица инцидентности;

множество ребер, упорядоченное по возрастанию весов;

множество ребер, упорядоченное по убыванию весов;

матрица весов.

10.Тренинговые задания

В качестве тренинговых заданий рекомендуется использовать все упражнения лабораторного практикума.

11.Перечень вопросов итоговой аттестации по курсу

Типовые вопросы для итогового контроля знаний:

1. Неориентированный граф. Определение. Смежность ребер и вершин. Инцидентность. Изоморфизм.
2. Теорема о четности вершин в конечном графе. Определения полного графа и подграфа.
3. Определение маршрута, замкнутого маршрута. Определение цепи и простой цепи. Определение дерева. Теорема о количестве ребер в дереве с n вершинами.
4. Связный неориентированный граф. Теорема о связности графа.
5. Определение орграфа. Смежность в орграфе. Положительная и отрицательная инцидентность в орграфе. Положительная и отрицательная степени вершин в орграфе.
6. Определение пустого графа. Определение изолированной вершины. Определение ормаршрута и замкнутого

- ормаршрута. Определение пути и простого пути. Определение контура.
7. Определение сильносвязного графа. Определение орграфа. Определение симметричного и смешанного графов.
 8. Определение эксцентриситета, диаметра и радиуса в графе. Центр графа.
 9. Матрица смежности и инцидентности для неорграфа. Список смежности. Матрица весов.
 10. Матрица смежности и инцидентности для орграфа.
 11. Теорема о числе ормаршрутов длины n между двумя вершинами орграфа.
 12. Построение минимального покрывающего дерева по алгоритму Краскала. Алгоритм.
 13. Построение максимального покрывающего дерева по алгоритму Краскала. Алгоритм.
 14. Определение псевдографа и мультиграфа. Определение плоского графа.
 15. Построение минимального покрывающего дерева по алгоритму Прима. Алгоритм.
 16. Построение максимального покрывающего дерева по алгоритму Прима. Алгоритм.
 17. Поиск маршрута и наименьшей длины по алгоритму Дейкстры. Алгоритм.
 18. Задача о кенигсбергских мостах. Определение эйлерова цикла и эйлерова графа. Определение эйлерова пути и способ получения эйлерова пути из эйлерова цикла.
 19. Теорема о четности степеней в эйлеровом графе.
 20. Поиск эйлерова цикла в графе. Алгоритм.
 21. Поиск минимальных расстояний между всеми парами вершин по алгоритму Уоршалла-Флойда. Алгоритм.
 22. Сравнение алгоритмов Дейкстры и Уоршалла-Флойда. Сходства и различия алгоритмов.
 23. Задача построения транзитивного замыкания бинарного отношения. Определение бинарного отношения. Определение транзитивного замыкания. Матрица достижимости (связности).
 24. Построение транзитивного замыкания для графа.

III. УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО ДИСЦИПЛИНЕ

Направление подготовки

080500 «Бизнес-информатика»

Квалификация (степень) выпускника бакалавр

1. ПРОГРАММА ДИСЦИПЛИНЫ

1. Цели и задачи дисциплины

Основной целью освоения дисциплины является изучение классической теории конечных графов, а также применение методов теории конечных графов в прикладных задачах. Курс «Теория конечных графов» носит теоретический и практический характер.

Задачей дисциплины является развитие навыков формализации и описания математических объектов, оперирование основными характеристиками графов и применение алгоритмов по основным темам дисциплины.

2. Место дисциплины в структуре ООП:

(указывается цикл, к которому относится дисциплина; формулируются требования к входным знаниям, умениям и компетенциям студента, необходимым для ее изучения; определяются дисциплины, для которых данная дисциплина является предшествующей)

Цикл, к которому относится дисциплина: вариативная часть профессионального цикла Б.3.

Требования к входным знаниям и умениям: необходима математическая подготовка в пределах школьной программы.

Знать Основные понятия и методы теории множеств.

Уметь: Анализировать выводы, полученные при решении задач.

Дисциплины, для которых данная дисциплина является предшествующей: Модели на гиперграфах, Введение в управление инфокоммуникациями, Проектирование корпоративных систем, Прикладные задачи ТМО.

3. Требования к результатам освоения дисциплины:

Процесс изучения дисциплины «Теория конечных графов» направлен на формирование следующих компетенций:
ОК: 1, ПК: 19, 20

(указываются в соответствии с ФГОС ВПО)

- ▲ владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановке цели и выбору путей ее достижения (ОК-1)
- ▲ использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования (ПК-19)
- ▲ готовить научно-технические отчеты, презентации, научные публикации по результатам выполненных исследований (ПК-20)

В результате изучения дисциплины «Теория конечных графов» студент должен:

Знать:

- концепции дисциплины: Теория конечных графов.
- основные законы теоретического исследования.

Уметь:

- использовать основные законы теоретического исследования; решать прикладные задачи по дисциплине «Теория конечных графов»,
- применять методы моделирования, теоретического и экспериментального исследования,
- понимать и применять в исследовательской и прикладной деятельности современный математический аппарат

Владеть:

- современным математическим аппаратом;

- вычислительными средствами;
- базовыми математическими знаниями.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 4 зачетных единиц.

Вид учебной работы	Всего часов	Семестры	
		3	
		3	
Аудиторные занятия (всего)	54	54	–
В том числе:			–
Лекции	36	36	–
Практические занятия (ПЗ)	–	–	–
Семинары (С)	–	–	–
Лабораторные работы (ЛР)	18	18	–
Самостоятельная работа (всего)	90	90	–
В том числе:	–	–	–
Курсовой проект (работа)	–	–	–
Расчетно-графические работы	–	–	–
Реферат	–	–	–
<i>Другие виды самостоятельной работы</i>	–	–	–
Самостоятельная проработка дополнительного материала	90	90	–
Вид промежуточной аттестации (зачет, экзамен)		экзамен	–
Общая трудоемкость, час	144	144	–
зач. ед.	4	4	–

5. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Элементы теории графов	Введение в теорию графов: основные понятия и определения. Матричные представления графов. Маршруты, цепи, циклы. Нахождение связных компонент. Метрические характеристики графов. Подграфы. Операции над графами. Двудольные графы. Поиск в ширину. Деревья. Эйлеровы графы. Гамильтоновы графы. Эйлеровы пути и циклы. Гамильтоновы пути и циклы. Связь между наличием в связном графе гамильтоновых циклов и длиной максимальных простых путей в нем. Нахождение кратчайших путей в ориентированном графе.
2.	Алгоритмы на графах	Алгоритм Краскала. Алгоритм Прима. Алгоритм Дейкстры. Алгоритм нахождения эйлерова цикла в графе. Алгоритм построения кратчайшего пути от фиксированной вершины до всех остальных вершин в ориентированном графе, случай неотрицательных весов ребер.
3.	Потоки в сетях	Прикладные модели и задачи, примеры применения методов теории графов. Оценки структурных компонент графа. Задача о максимальном потоке и о

		минимальном разрезе в сети. Максимальный поток в транспортной сети. Задача на нахождение «узких» мест в сети. Задача о потоке минимальной стоимости.
--	--	--

5.1. Содержание разделов дисциплины

5.2 Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин		
		1	2	3
1.	Модели на гиперграфах	+	+	+
2.	Введение в управление инфо-коммуникациями	+	+	+
3.	Моделирование информационных процессов			+
4.	Прикладные задачи ТМО	+		+
5.	Курсовая работа	+	+	+

5.3. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	Семина	СРС	Всего час.
1.	Элементы теории графов	12	0	6	0	30	48
2.	Алгоритмы на графах	12	0	6	0	30	48
3.	Потоки в сетях	12	0	6	0	30	48
		36		18		90	144

6. Лабораторный практикум

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час.)
1.	Элементы теории графов	Разбор основных понятий и определений на неориентированных и ориентированных графах. Матрицы смежности и матрицы инцидентности для неориентированных и ориентированных графов. Поиск маршрутов, цепей и циклов для графов. Нахождение связных компонент в графе. Метрические характеристики графов. Эйлеровы графы. Эйлеровы	6

		пути и циклы. Нахождение кратчайших путей в ориентированном графе	
2.	Алгоритмы на графах	Алгоритм Краскала. Алгоритм Прима. Алгоритм Дейкстры. Алгоритм нахождения эйлера цикла в графе. Алгоритм построения кратчайшего пути от фиксированной вершины до всех остальных вершин в ориентированном графе, случай неотрицательных весов ребер. Построение матрицы связности (достижимости для графа). Алгоритм Уоршалла-Флойда.	6
3.	Потоки в сетях	Прикладные модели и задачи. Оценка структурных компонент графа. Задача о максимальном потоке и о минимальном разрезе в сети. Максимальный поток в транспортной сети. Задача на нахождение «узких» мест в сети. Задача о потоке минимальной стоимости.	6
			18

7. Практические занятия (семинары)

не предусмотрены

8. Примерная тематика курсовых проектов (работ)

не предусмотрены

9. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература

1. Гайдамака Ю.В., Зарипова Э.Р., Кокотчикова М.Г., Севастьянов Л.А. «Лекции по дискретной математике. Часть II. Комбинаторика. Теория конечных графов». Учебное пособие. // М.: Изд-во РУДН, 2008.
2. Харари Фрэнк. Теория графов / Харари Фрэнк ; Пер. с англ. В.П.Козырева; Под ред. Г.П.Гаврилова. – 4-е изд. – М. : URSS : Либроком, 2009. – 296 с.
3. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. «Лекции по теории графов». Изд.2, испр., Изд-во Либроком, 2009.
4. Муромцев В.В. «Некоторые алгоритмы на графах». Учебное пособие, Белгород: Изд. БелГТАСМ, 2000.– 64 с.

б) дополнительная литература

1. Зыков А. А. Основы теории графов. — М.: «Вузовская книга», 2004. — С. 664
2. Кирсанов М.Н. «Графы в MAPLE» Пособие по дискретной математике для студентов университетов. М: Физматлит, 2007.

в) программное обеспечение Maple, MatLab, SciLab, MathCad, Mathematica

г) базы данных, информационно-справочные и поисковые системы

<http://vuz.exponenta.ru/PDF/book/GrMaple.pdf>

10. Материально-техническое обеспечение дисциплины

Москва, ул. Орджоникидзе, д.3, корп. 1. учебные лаборатории кафедры систем телекоммуникаций:

1. ауд. 110: проектор DMS800 с интерактивной доской Board 1077, ноутбук Toshiba Satellite 17/300GB Intel Core2 2.4 GHz (10 шт.)
2. ауд. 114: проектор DMS800 с интерактивной доской Board 1077 ноутбук Toshiba Satellite 17/300GB Intel Core2 2.4 GHz (10 шт.)
3. ауд. 116: проектор DMS800 с интерактивной доской Board 1077, HP xw7800, Intel Core2 2.4 GHz (8 шт.)

11. Методические рекомендации по организации изучения дисциплины

На освоение дисциплины отводится 1 семестр. В качестве итогового контроля знаний предусмотрен экзамен.

Для текущего контроля успеваемости и промежуточной аттестации студентов рекомендуется использовать вопросы и задания подобные перечисленным ниже:

Типовые задачи для промежуточного контроля знаний:

1. Найти эксцентриситет, диаметр и радиус графа.
2. Составить матрицу смежности и матрицу инцидентности для графа.
3. Построить минимальное покрывающее дерево для графа по алгоритму Краскала.
4. Задача на применение алгоритма Дейкстры.
5. Задача на применение алгоритма Уоршалла-Флойда.
6. Поиск эйлера цикла в графе.

Типовые вопросы для итогового контроля знаний:

1. Неориентированный граф. Определение. Смежность ребер и вершин. Инцидентность. Изоморфизм.
2. Теорема о четности вершин в конечном графе. Определения полного графа и подграфа.
3. Определение маршрута, замкнутого маршрута. Определение цепи и простой цепи. Определение дерева.

- Теорема о количестве ребер в дереве с n вершинами.
4. Связный неориентированный граф. Теорема о связности графа.
 5. Определение орграфа. Смежность в орграфе. Положительная и отрицательная инцидентность в орграфе. Положительная и отрицательная степени вершин в орграфе.
 6. Определение пустого графа. Определение изолированной вершины. Определение ормаршрута и замкнутого ормаршрута. Определение пути и простого пути. Определение контура.
 7. Определение сильносвязного графа. Определение орграфа. Определение симметричного и смешанного графов.
 8. Определение эксцентриситета, диаметра и радиуса в графе. Центр графа.
 9. Матрица смежности и инцидентности для неорграфа. Список смежности. Матрица весов.
 10. Матрица смежности и инцидентности для орграфа.
 11. Теорема о числе ормаршрутов длины n между двумя вершинами орграфа.
 12. Построение минимального покрывающего дерева по алгоритму Краскала. Алгоритм.
 13. Построение максимального покрывающего дерева по алгоритму Краскала. Алгоритм.
 14. Определение псевдографа и мультиграфа. Определение плоского графа.
 15. Построение минимального покрывающего дерева по алгоритму Прима. Алгоритм.
 16. Построение максимального покрывающего дерева по алгоритму Прима. Алгоритм.
 17. Поиск маршрута и наименьшей длины по алгоритму Дейкстры. Алгоритм.
 18. Задача о кенигсбергских мостах. Определение эйлерова цикла и эйлерова графа. Определение эйлерова пути и способ получения эйлерова пути из эйлерова цикла.
 19. Теорема о четности степеней в эйлеровом графе.
 20. Поиск эйлерова цикла в графе. Алгоритм.

21. Поиск минимальных расстояний между всеми парами вершин по алгоритму Уоршалла-Флойда. Алгоритм.
22. Сравнение алгоритмов Дейкстры и Уоршалла-Флойда. Сходства и различия алгоритмов.
23. Задача построения транзитивного замыкания бинарного отношения. Определение бинарного отношения. Определение транзитивного замыкания. Матрица достижимости (связности).
24. Построение транзитивного замыкания для графа. Алгоритм.
25. Особенности i -той строки и i -столбца для Алгоритма Уоршалла-Флойда. Доказательство.
26. Особенности i -той строки и i -столбца для Алгоритма поиска транзитивного замыкания. Доказательство.
27. Определение потока в графе. Условия существования потока в графе. Правила раскрашивания дуг графа для поиска увеличивающей цепи. Определение прямых и обратных дуг.
28. Увеличение потока в графе по увеличивающей цепи. Алгоритм.
29. Поиск максимального потока в графе. Алгоритм.
30. Поиск потока минимальной стоимости. Алгоритм.
31. Поиск оптимального маршрута почтальона для орграфа. Алгоритм.
32. Определение гамильтонова цикла и гамильтоновой цепи.
33. Гамильтоновы и эйлеровы графы. Сходства и различия гамильтонова и эйлерова циклов.
34. Три достаточных критерия существования гамильтоновых циклов в неорграфе.
35. Поиск гамильтонова цикла в орграфе. Алгоритм с упрощением

ЛИТЕРАТУРА

1. Иванов Б.Н. «Дискретная математика. Алгоритмы и программы». // М.: Изд-во «Лаборатория базовых знаний», 2003.
2. Харари Ф. «Теория графов». // М.: "Мир", 2007.
3. Ю.В. Гайдамака, К.Е. Самуйлов, Л.А. Севастьянов, С.С. Спесивов «Комбинаторика. Алгоритмы на графах». Учебно-методическое пособие. // М.: Изд-во РУДН, 2002.
4. Г.П. Гаврилов, А.А. Сапоженко «Сборник задач по дискретной математике». // М.: Наука, 2007.
5. Муромцев В.В. Некоторые алгоритмы на графах : Учебное пособие. Белгород: Изд. БелГТАСМ, 2000.- 64 с.

Оглавление

I. КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ	3
Тема 1. Графы. Неориентированные графы: основные понятия; маршруты, цепи, циклы; связность; деревья и леса.....	3
Тема 2. Ориентированные графы: основные понятия; ориентированные маршруты, пути, контуры; сильная связность. Ориентированные деревья	16
Тема 3. Метрические характеристики графов. Матричное представление графов: матрица инцидентности для неорграфа, матрица смежности для неорграфа, матрица инцидентности для орграфа, матрица смежности для орграфа. Список смежности.....	27
Тема 4. Построение покрывающих деревьев. Алгоритм Краскала. Построение покрывающего дерева для связного графа. Построение минимального покрывающего дерева по алгоритму Краскала. Построение максимального покрывающего дерева по алгоритму Краскала	37
Тема 5. Построение минимального покрывающего дерева для связного взвешенного графа по алгоритму Прима. Построение максимального покрывающего дерева по алгоритму Прима.....	50
Тема 6. Поиск пути наименьшей длины в графе. Алгоритм Дейкстры ...	56
Тема 7. Эйлеровы графы. Алгоритм поиска эйлерова цикла в графе	61
Тема 7. Гамильтоновы графы. Сходство и различия гамильтоновых и эйлеровых графов. Достаточные условия существования гамильтоновых циклов. Способы поиска гамильтонова цикла. Алгоритм поиска гамильтонова цикла в графе.....	69
Тема 9. Поиск расстояния между всеми парами вершин. Алгоритм Уоршалла-Флойда.....	79
Тема 10. Задача построения транзитивного замыкания бинарного отношения. Алгоритм построения транзитивного замыкания бинарного отношения.....	82
Тема 11. Потoki. Условия существования потока. Увеличивающая цепь. Алгоритм поиска увеличивающей цепи. Увеличение потока вдоль найденной цепи по правилам	86
Тема 12. Потoki. Поиск максимального потока. Поиск потока минимальной стоимости.....	94
Тема 13. Задача почтальона для орграфов. Алгоритм поиска	

оптимального маршрута почтальона для оргграфов	101
II. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ	107
1. Словарь (гlossарий) основных терминов и понятий.....	107
2. Методические указания для преподавателя, студента, слушателя 111	
3. Сборник задач и упражнений	111
4. Лабораторный практикум по дисциплине	112
5. Описание балльно-рейтинговой системы	145
6. Вопросы для самопроверки и обсуждений по темам.....	147
7. Задания для самостоятельной работы по темам.....	148
8. Перечень рефератов и/или курсовых работ по темам.....	151
9. Тестовые задания по темам (для текущего и промежуточного самоконтроля).....	151
10. Тренинговые задания.....	159
11. Перечень вопросов итоговой аттестации по курсу	159
III. УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО ДИСЦИПЛИНЕ.....	162
ЛИТЕРАТУРА	173

Учебное издание

Эльвира Ринатовна Зарипова
Мария Геннадьевна Кокотчикова

Дискретная математика
Часть III. Теория графов

Учебное пособие

Издание подготовлено в авторской редакции

Технический редактор Н.А.Ясько

Тематический план 2013 г. №12

Подписано в печать 07.08.12

Усл.печ.л. 4,65

Тираж 200, Заказ 370

Российский университет дружбы народов
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

Типография ИПК РУДН
ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41