Московский государственный технический университет
имени Н. Э. Баумана

Т. И. Бояринцева, А. А. Мастихина

Теория графов

*Методические указания к выполнению домашнего задания
по курсу «Дискретная математика»*

ИЗДАТЕЛЬСТВО
МГТУ им. Н.Э. БАУМАНА

Москва

2014

УДК 519.17
ББК 22.176
Б86

Издание доступно в электронном виде на портале *ebooks.bmstu.ru*
по адресу: http://ebooks.bmstu.ru/catalog/109/book218.html

Факультет «Фундаментальные науки»

Кафедра «Высшая математика»

*Рекомендовано Учебно-методической комиссией
Научно-учебного комплекса «Фундаментальные науки»
МГТУ им. Н. Э. Баумана*

Рецензент: канд. физ.-мат. наук, доцент *О. В. Пугачев*

Бояринцева Т. И.

Б86 Теория графов : метод. указания / Т. И. Бояринцева, А. А. Мастихина. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2014. — 37, [3] с. : ил.

ISBN 978-5-7038-3994-2

Изложены основные понятия и теоретические результаты применения теории графов. Приведены примеры, рассмотрены типовые задачи.

Для студентов факультета «Робототехника и комплексная автоматизация», изучающих курс «Дискретная математика».

УДК 519.17
ББК 22.176

ISBN 978-5-7038-3994-2

ПРЕДИСЛОВИЕ

Данные методические указания предназначены для студентов МГТУ им. Н.Э. Баумана, выполняющих типовой расчет по теории графов. Задачи типового расчета приведены в конце работы.

Теория графов является разделом дискретной математики. С помощью графов решается ряд оптимизационных задач. Указания написаны по материалам лекций и семинаров курса «Дискретная математика», читаемого авторами в МГТУ им. Н.Э. Баумана.

В методических указаниях изложены основные понятия теории графов, рассмотрены типовые задачи, которые решаются средствами данной теории, и представлены алгоритмы их решения.

1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПОНЯТИЯ

Графом называют совокупность элементов (*вершин* графа) $V = \{v_1, v_2, \ldots, v_n\}$ и связей между ними $E = \{e_1, e_2, \ldots, e_m\}$ (*ребер*), причем каждое ребро e_k является парой $\{v_i, v_j\}$ различных элементов из множества V, обозначая наличие связи между ними, т. е. это пара конечных множеств $G = (V, E)$, где каждый элемент E — пара вершин из множества V.

Ориентированным графом (орграфом) называют пару $G = (V, E)$ множества *вершин* $V = \{v_1, v_2, \ldots, v_n\}$ и множества *дуг* $E = \{e_1, e_2, \ldots$ $\ldots, e_m\}$, где каждая дуга e_k является упорядоченной парой (v_i, v_j) различных элементов из множества V.

По умолчанию графом называют неориентированный граф.

Граф называют *взвешенным*, если каждому ребру приписано некоторое число, обозначающее длину, вес или какую-либо другую величину.

Поставим в соответствие вершинам графа точки на плоскости или в трехмерном пространстве, а каждому ребру — линию, соединяющую точки, соответствующие вершинам, каждой дуге — такую же линию, но с указанием направления (стрелкой). Получившееся изображение будем называть *геометрическим графом*.

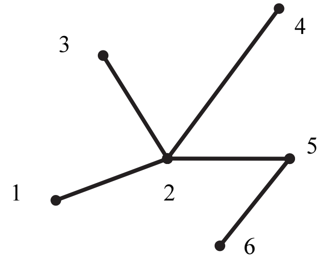


Рис. 1

Граф $G = (V, E)$, $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{5, 6\}\}$, изображен на рис. 1.

Два графа называют *изоморфными*, если существует взаимно-однозначное отображение множеств их вершин, сохраняющее связи между ними. Ясно, что различные геометрические графы могут быть изоморфны одному графу: геометрический граф, показанный на рис. 2, также соответствует графу G.

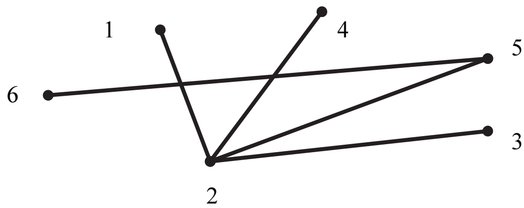


Рис. 2

Далее часто графы будут задаваться в геометрическом виде.

Две соединенные между собой вершины называют *смежными* (т. е. $\{v, u\} \in E$).

Ребро e называют *инцидентным* вершине v, если $e = \{v, u\}$, $u \in V$.

Граф $G_1 = (V_1, E_1)$ называют *подграфом* графа $G = (V, E)$ (обозначают $G_1 \subseteq G$), если $V_1 \subseteq V$ и $E_1 \subseteq E$.

Степенью вершины $d(v)$ неориентированного графа называют число ребер, инцидентных этой вершине: $d(v) = |\{u: \{u, v\} \in E\}|$.

Пример графа с указанием степеней его вершин показан на рис. 3. Здесь $d(v_1) = 1$, $d(v_2) = 2$, $d(v_3) = 2$, $d(v_4) = 2$, $d(v_5) = 4$, $d(v_6) = 3$.

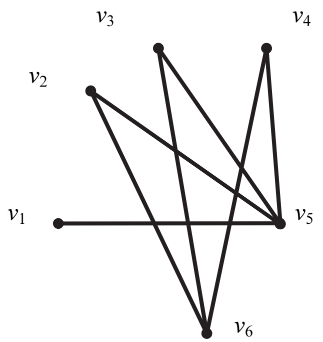


Рис. 3

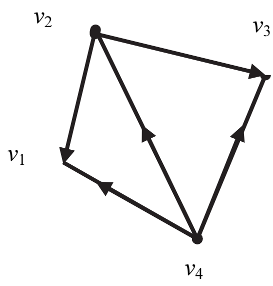


Рис. 4

Для ориентированных графов определены понятия полустепени захода $d_+(v)$ и полустепени исхода $d_-(v)$ — как число дуг, входящих в вершину $d_+(v) = |\{u\colon (u, v) \in E\}|$ и количество выходящих из нее дуг, $d_-(v) = |\{u\colon (v, u) \in E\}|$. Пример ориентированного графа с указанием полустепеней исхода и полустепеней захода рассмотрен на рис. 4. Здесь $d_+(v_1) = 0$, $d_-(v_1) = 2$, $d_+(v_2) = 2$, $d_-(v_2) = 1$, $d_+(v_3) = 0$, $d_-(v_3) = 2$, $d_+(v_4) = 3$, $d_-(v_4) = 0$.

Графы без ребер (дуг) называют *пустыми* и обозначают O_n, где n — число вершин. Примеры таких графов показаны на рис. 5.

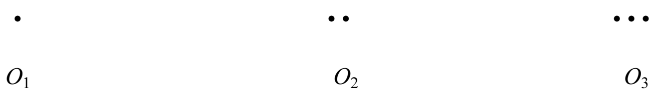


Рис. 5

Графы с n вершинами, в которых любая вершина связана со всеми остальными вершинами, называют *полными* и обозначают K_n (рис. 6).

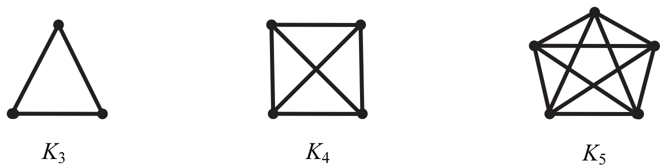


Рис. 6

Циклы — графы, в которых вершины соединены последовательно, обозначают C_n. Самый маленький цикл — это C_3. Примеры циклов приведены на рис. 7.

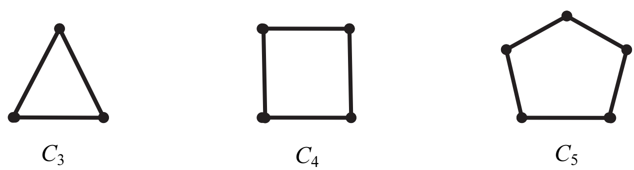


Рис. 7

Двудольными называются графы, в которых можно выделить два множества вершин — V_1 и V_2, две доли, так, что вершины одной доли не смежны между собой, но могут быть смежны с какими-либо вершинами другой доли. Пример двудольного графа показан на рис. 8.

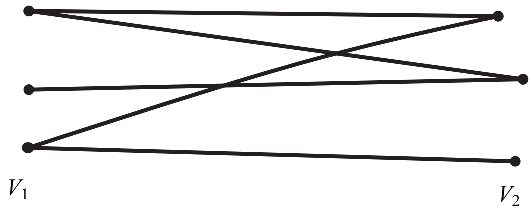


Рис. 8

Среди двудольных графов выделяют *полные двудольные* — те, у которых все вершины одной доли соединены с каждой вершиной другой доли. Их обозначают $K_{n,m}$, где $n = |V_1|$, $m = |V_2|$.

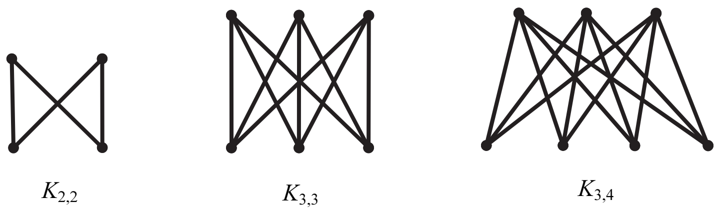


Рис. 9

Звезды — полные двудольные графы, в которых одна доля состоит из единственной вершины, $K_{n,1}$. На рис. 10 показаны звезды с тремя и пятью лучами.

Рис. 10

Колеса — графы, которые можно представить как звезду, вписанную в цикл. Их обозначают W_n. Примеры колес даны на рис. 11.

Рис. 11

Платоновыми телами называют правильные многогранники, т. е. такие, в которых все грани одинаковы. Всего существует пять платоновых тел. На рис. 12 – 16 слева показаны тела, а справа — их графы.

• Тетраэдр (4 вершины, 6 ребер, 4 грани — треугольники, рис. 12).

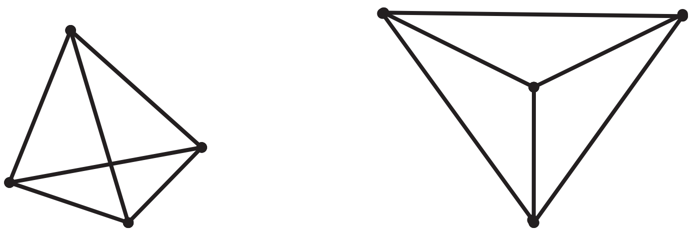


Рис. 12

• Куб, 8 вершин (12 ребер, 6 граней — четырехугольники, рис. 13).

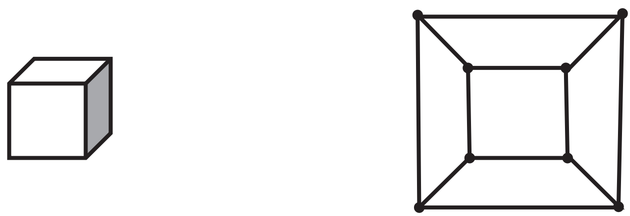


Рис. 13

• Октаэдр (6 вершин,12 ребер, 8 граней — треугольники, рис. 14).

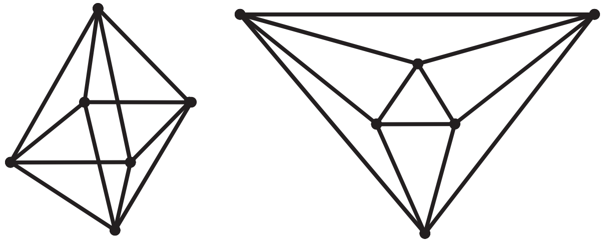


Рис. 14

• Додекаэдр (12 граней — пятиугольники, рис. 15).

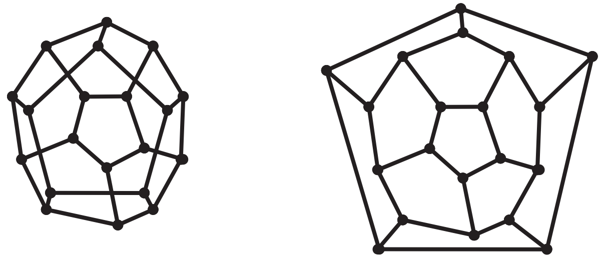


Рис. 15

• Икосаэдр (20 граней — треугольники, рис. 16).

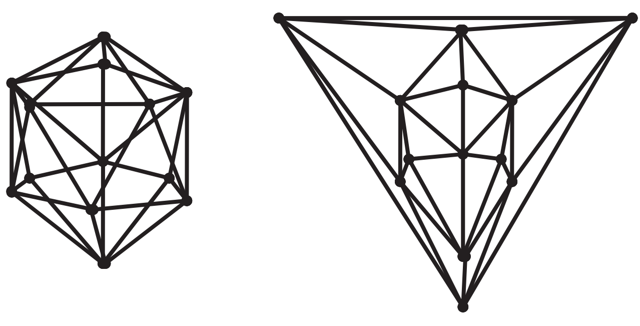


Рис. 16

Отметим, что хотя платоновы тела являются правильными многогранниками (стороны равны), соответствующие им графы обозначают только вершины и связи между ними.

Мультиграфы — графы, в которых между двумя вершинами может быть больше одной связи (рис. 17). Мультиграфы могут быть ориентированными и неориентированными.

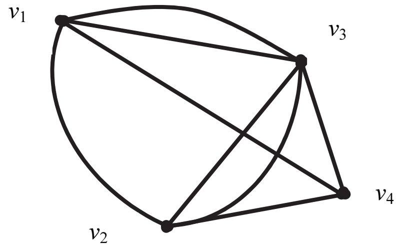


Рис. 17

Псевдографы — мультиграфы, в которых также допустимы петли — ребра или дуги, соединяющие одну и ту же вершину. Пример псевдографа дан на рис. 18.

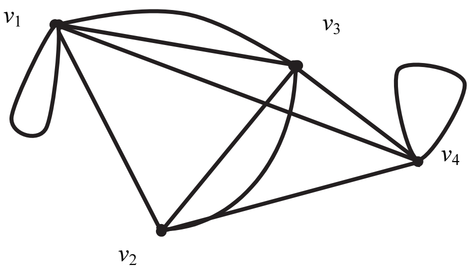


Рис. 18

Проекцией ориентированного графа называют неориентированный граф, полученный из исходного путем снятия ориентации (то есть вершины u,v в проекции соединяются ребром, если в ориентированном графе есть дуга (u,v), (v,u) или обе).

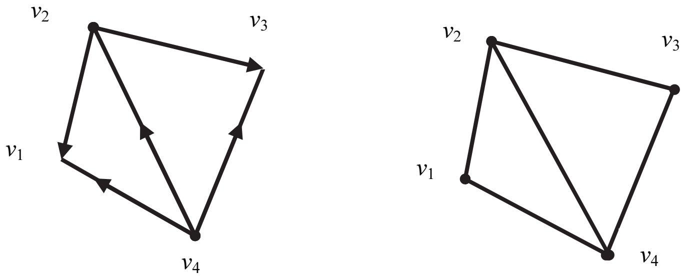


Рис. 19

Пример проекции приведен на рис. 19.

2. СПОСОБЫ ЗАДАНИЯ ГРАФА

Граф G — два множества (V, E) — может быть задан *словесным описанием*, обозначающим связи между вершинами, *с помощью изображения* (геометрически), а также следующими способами:

• *матрицей смежности* $S(G)$ — матрицей размера $n \times n$ (n — число вершин графа), состоящей из нулей и единиц. Элемент $s_{ij} = 1$ тогда и только тогда, когда $\{v_i,\ v_j\} \in E$ или $(v_i, v_j) \in E$, если это ориентированный граф. На главной диагонали матрицы, таким образом, находятся нули. Матрица смежности для неориентированных графов является симметричной. Для мультиграфов вместо единиц в соответствующих ячейках матрицы приводят число ребер. Для псевдографов на диагонали указывают наличие и число петель;

• *списком смежности.* Для каждой вершины указывают список вершин, смежных с ней. Для ориентированных графов для u приводят список $\{v \in V,\ (u, v) \in E\}$;

• *матрицей инцидентности* — матрицей размера $n \times m$ (m — число ребер (дуг) графа), состоящей из нулей и единиц. Элемент $a_{ij} = 1$ тогда и только тогда, когда ребро e_j инцидентно вершине v_i.

Рассмотрим на конкретном примере различные способы задания графа.

Словесное описание. В двухкомнатной квартире помещения считаются соединенными между собой, если из одного в другое можно попасть за один шаг (рис. 20).

Пусть комнаты — вершины графа *G*, а ребра проводятся, если комнаты соединены. Это неориентированный граф. Числами от 1 до 8 обозначены ребра.

Геометрическое задание графа показано на рис. 21.

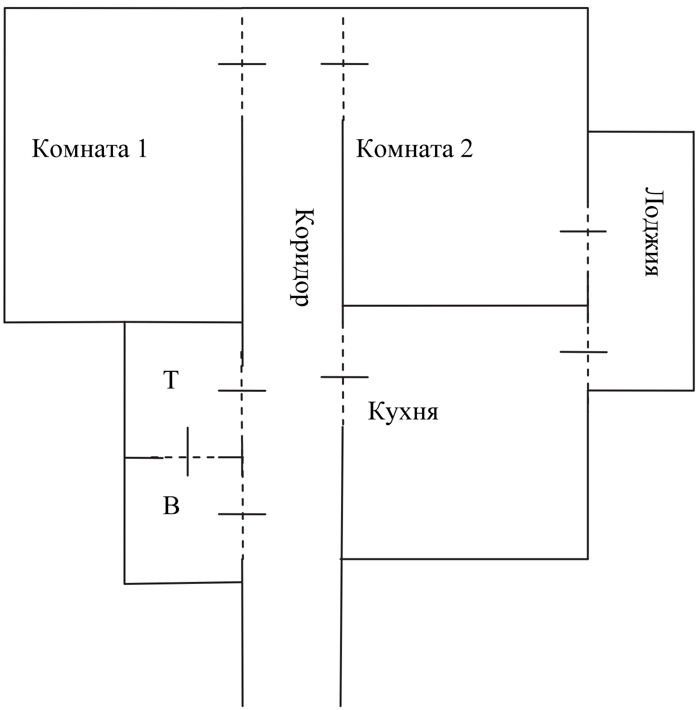


Рис. 20

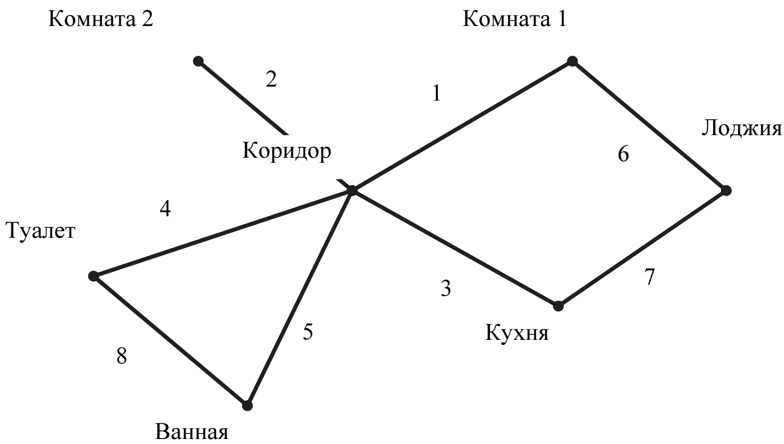


Рис. 21

Матрица смежности $S(G)$ этого графа имеет следующий вид:

	Коридор	Комната 1	Комната 2	Кухня	Лоджия	Ванная	Туалет
Коридор	1	1	1	1	0	1	1
Комната 1	1	1	0	0	1	0	0
Комната 2	1	0	1	0	0	0	0
Кухня	1	0	0	1	1	0	0
Лоджия	0	1	0	1	1	0	0
Ванная	1	0	0	0	0	1	1
Туалет	1	0	0	0	0	1	1

Списки смежности:

коридор — {комната 1, комната 2, кухня, ванная, туалет, коридор};
комната 1 — {коридор, лоджия, комната 1};
комната 2 — {коридор, комната 2};
кухня — {коридор, лоджия, кухня};
лоджия — {комната 1, кухня, лоджия};
ванная — {коридор, туалет, ванная};
туалет — {коридор, ванная, туалет}.

Матрица инцидентности имеет следующий вид:

e_j

		1	2	3	4	5	6	7	8
	Коридор	1	1	1	1	1	0	0	0
	Комната 1	1	0	0	0	0	1	0	0
	Комната 2	0	1	0	0	0	0	0	0
v_{i}	Кухня	0	0	1	0	0	0	1	0
	Лоджия	0	0	0	0	0	1	1	0
	Ванная	0	0	0	1	0	0	0	1
	Туалет	0	0	0	0	1	0	0	1

3. ЗАДАЧИ ОБ ОБХОДАХ ГРАФА

Путь из вершины v_1 в вершину v_k *в неориентированном графе* — чередующаяся последовательность вершин и ребер графа вида v_1, $\{v_1, v_2\}$, v_2, $\{v_2, v_3\}, \ldots, \{v_{k-2}, v_{k-1}\}$, v_{k-1}, $\{v_{k-1}, v_k\}$, v_k, причем все ребра различны.

Путь из вершины v_1 в вершину v_k *в ориентированном графе* — последовательность вершин и различных дуг графа вида v_1, (v_1, v_2), v_2, (v_2, v_3), …, (v_{k-1}, v_k), v_k.

Также можно записывать путь как последовательность одних вершин или одних ребер (дуг).

Путь называют *замкнутым*, если вершины v_1 и v_k совпадают.

Путь называют *простым*, если все вершины $v_1, v_2, v_3, \ldots, v_{k-1}, v_k$, кроме, может быть, v_1 и v_k, различны. Простой незамкнутый путь называют *цепью*.

Простой замкнутый путь называется *циклом*. Также цикл можно определять как подграф, изоморфный C_k.

Вершину v называют *достижимой* из вершины u, если существует путь из u в v.

Неориентированный граф, в котором все вершины достижимы друг из друга (т. е. из любой вершины, двигаясь по ребрам, можно попасть в любую другую вершину графа), называют *связным*.

Ориентированный граф называют связным, если связной является его проекция (как, например, граф, показанный на рис. 19).

Компонентой связности называется максимальный по включению связный подграф, т. е. такой подграф G_1, что не найдется другого связного подграфа G, содержащего G_1.

Ориентированный граф называют *сильно связным*, если в нем каждая вершина достижима из каждой.

Компонентой сильной связности ориентированного графа называют его максимальный по включению сильно связный подграф.

Мостом в теории графов называется ребро (дуга), при удалении которого(ой) теряется связность (увеличивается количество компонент связности). Пример графа, содержащего мост, показан на рис. 22 (ребра $\{v_2, v_5\}$, $\{v_4, v_5\}$ являются мостами).

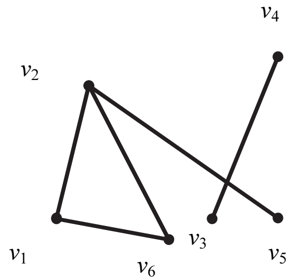


Рис. 22

Для определения количества и состава компонент связности используются алгоритмы обхода: поиск в глубину и поиск в ширину.

Поиск в глубину. Пусть вершины графа перенумерованы. Обход начинают из вершины с наименьшим номером v_1. Выбирают непройденную вершину с наименьшим номером, смежную с ней. (Если нет ни одной смежной вершины, обход связной компоненты завершен.)

Шаг алгоритма. Для каждой следующей проходимой вершины v_t выбирают непройденную вершину с наименьшим номером, смежную с ней. Она становится текущей на следующем шаге. Если такой нет, возвращаются в предыдущую пройденную вершину.

Шаг повторяют до тех пор, пока это возможно, т. е. пока не произойдет возврат в первую вершину v_1 и смежных с ней непройденных вершин не останется. Все пройденные вершины и соединяющие их ребра образуют компоненту связности.

Если пройдены не все вершины графа, граф несвязный. Следует взять вершину с наименьшим номером, которая не была пройдена, и повторять шаг, пока это возможно (т. е. провести обход следующей компоненты). И так до тех пор, пока все вершины не будут пройдены.

Результатом обхода будут последовательности вершин связных компонент, расположенных по пути обхода.

Для графа, показанного на рис. 21, поиск в глубину дает такой обход:

$v_1 \to v_2 \to v_5 \to v_2 \to v_6 \to v_2 \to v_1$;

$v_3 \to v_4 \to v_3$.

Поиск в ширину. В результате этого поиска вершины распределяются по уровням, характеризующим удаленность от первой вершины.

Возьмем вершину v_1. Это будет вершина нулевого уровня. Уровень $i+1$ образуется непройденными вершинами, смежными с вершинами i-го уровня.

Если уровень оказывается пустым, обход связной компоненты завершен. В случае если пройдены не все вершины, аналогично выполняют обход других компонент.

Для графа, изображенного на рис. 22, поиск в ширину дает следующий результат:

$$\begin{array}{ccccl}
 & v_1 & & v_3 & \textit{уровень}\ 0\\
\swarrow & & \searrow & \downarrow & \\
v_2 & & v_6 & v_4 & \textit{уровень}\ 1\\
\downarrow & & & & \\
v_5 & & & & \textit{уровень}\ 2
\end{array}$$

Определять достижимость вершин можно также с помощью операций с матрицей смежности.

Стабилизация матрицы. Рассмотрим матрицу смежности S графа G, но на главной диагонали поставим единицы. Ее можно считать матрицей достижимости за один шаг — на пересечении i-й строки и j-го столбца стоит единица, если i-я и j-я достижимы друг из друга не более чем по одному ребру (т. е. смежны или совпадают). S^2 — матрица достижимости не более чем за два шага, эта матрица получается возведением матрицы S в квадрат, причем, если при перемножении i-й строки и j-го столбца получается число, большее единицы, то все равно ставим единицу. Аналогично определяются матрицы S^3, S^4 и т. д. Если наступает момент, когда матрица достижимости за n шагов равна матрице достижимости за $(n-1)$ шаг, говорят о наступлении стабилизации матрицы, эту матрицу называют матрицей достижимости. $D = S^{n-1} = S^n$. Если в матрице достижимости на пересечении i-й строки и j-го столбца стоит единица, это означает, что j-я вершина может быть достигнута из i-й за конечное число шагов.

Заметим, что в матрице достижимости для неориентированного графа всем взаимно достижимым вершинам будут соответствовать одинаковые строки. Таким образом, если граф имеет k компонент связности, то в матрице достижимости будет ровно k групп одинаковых строк, и каждая такая группа будет соответствовать своей компоненте связности.

Задача о кенигсбергских мостах. Кенигсберг (современное название — Калининград) стоит на реке Прегель. Карта города изображена на рис. 23. Ее схематичное изображение дано на рис. 24.

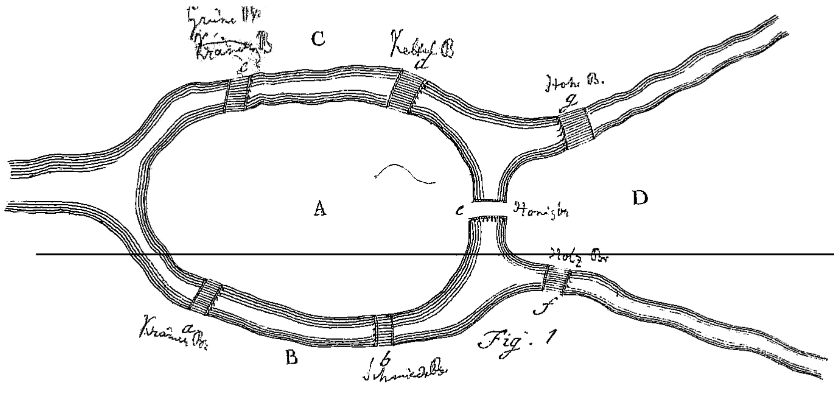


Рис. 23

Издавна была известна головоломка: нужно пройти по всем мостам по одному разу и вернуться в исходное место. Однако никому это не удавалось. Пройти по всем мостам по одному разу, даже не возвращаясь на прежнее место, — тоже. Данная задача в теории графов получила название «задача о кенигсбергских мостах». В 1736 г. Эйлер опубликовал общее решение таких задач.

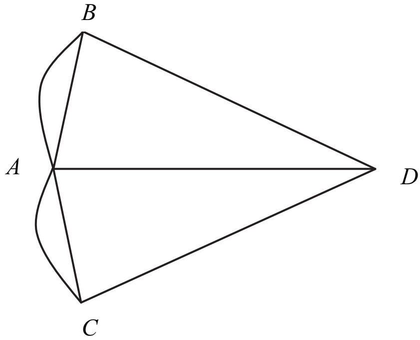


Рис. 24

Теорема Эйлера. В связном графе можно обойти все ребра ровно по одному разу и вернуться в исходную вершину тогда и только тогда, когда все вершины четны. Такой граф называется *эйлеровым*, а замкнутый путь, содержащий все ребра — эйлеровым циклом.

В связном графе можно обойти все ребра ровно по одному разу, но не вернуться в исходную вершину, тогда и только тогда, когда в нем ровно две вершины нечетны. Такой граф называют *полуэйлеровым*, а путь, содержащий все ребра, — *эйлеровой цепью*. В противном случае обход всех ребер только по одному разу невозможен.

В детских журналах часто публикуют задачи типа «нарисуй, не отрывая карандаша». Если считать, что точки, в которых сходятся несколько линий — это вершины графа, а сами линии — ребра, подобные рисунки можно рассматривать как графы. Эти задачи попадают под действие теоремы Эйлера и решаются с помощью алгоритма Флери.

Алгоритм Флери:

1) если граф эйлеров, то начинают движение из любой вершины, если он полуэйлеров, то из одной из нечетных. Путь должен закончиться в той же вершине либо во второй нечетной;

2) из каждой вершины проходят по любому инцидентному ребру, только если оно не является мостом. Пройденные ребра за собой стирают.

Алгоритм заканчивает работу, когда ребер больше нет.

В алгоритме Флери перед началом поиска решения с помощью алгоритмов обхода или стабилизации матрицы определяют начальное число компонент связности. Если их больше одной, то граф несвязный, а значит, решить задачу о кенигсбергских мостах для него в принципе невозможно. Если компонента одна, переходят к п. 1, иначе — к п. 2. Выбирают ребро и пробуют его удалить, анализируют полученное число компонент связности. Если оно возросло, то восстанавливают удаленное ребро и ищут другое из той же вершины. В таком модифицированном виде алгоритм Флери годится для реализации на ЭВМ.

Задача коммивояжера. В 1859 г. ирландским математиком У. Гамильтоном была предложена игра «Кругосветное путешествие». Каждой из двадцати вершин додекаэдра приписано название одного из крупных городов мира. Коммивояжер, переходя из одного города в другой по ребрам додекаэдра, должен посетить все города по одному разу и вернуться в исходный город. То есть построен граф в виде додекаэдра, и требуется, двигаясь по ребрам, обойти все его вершины по одному разу и вернуться в исходную вершину. Обобщив задачу на все графы, получаем современную постановку задачи коммивояжера — отыскать в графе цикл, содержащий все вершины и только по одному разу. Такой цикл называют *гамильтоновым*. Граф, в котором есть гамильтонов цикл, также называют гамильтоновым. Бывает, что гамильтонов цикл выделить нельзя, но можно построить незамкнутую цепь, содержащую все вершины по одному разу. Ее тоже называют гамильтоновой.

Теорема Дирака (1952 г.). Если число вершин связного графа не меньше 3 ($n \geq 3$) и степень каждой вершины больше или равна половине числа вершин ($d(v) \geq n/2$), то граф является гамильтоновым.

Эта теорема дает достаточное условие того, что граф гамильтонов, но оно не является необходимым, т. е. если условия теоремы Дирака не выполняются, это еще не означает, что в нем нельзя обойти все вершины по одному разу и вернуться в исходную точку. Например, граф C_6: степень каждой вершины равна 2, а половина всех вершин — 3, условие теоремы нарушено, однако граф является гамильтоновым.

Алгоритм с откаткой (поиск в глубину). Начинают движение из произвольной вершины. Двигаемся по ребрам произвольно, не заходя дважды в одну вершину. Если дальнейшее продвижение невозможно, делают шаг назад, при этом вершину v, из которой произошел возврат, снова считают непройденной, а ребро $\{u, v\}$, по которому вернулись, — отсутствующим, пока u считается пройденной. Продолжают поиски вариантов продвижения. Алгоритм заканчивает работу, когда пройдены все вершины, то есть построен гамильтонов цикл/цепь, либо когда произошел возврат в исходную вершину, и из нее больше нет выходов.

4. ДЕРЕВО. МИНИМАЛЬНОЕ ОСТОВНОЕ ДЕРЕВО

Деревом называют связный граф без циклов. Пример дерева приведен на рис. 25.

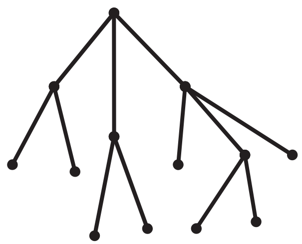


Рис. 25

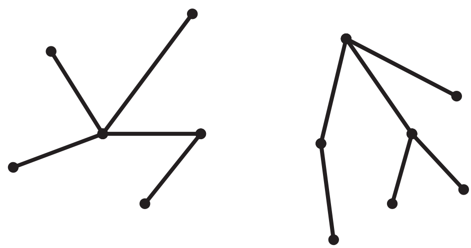


Рис. 26

Граф без циклов, состоящий из нескольких компонент связности, называют *лесом.* Очевидно, что каждая компонента связности леса является деревом. Пример леса показан на рис. 26.

Деревья обладают следующими свойствами:

1) в дереве число ребер на единицу меньше числа вершин;

2) если добавить к дереву любое ребро, не добавляя вершин, в графе возникнет цикл;

3) если удалить из дерева любое ребро, не удаляя вершин, граф перестанет быть связным.

Пусть есть связный граф $G = (V, E)$. В нем можно выделить различными способами подграф $F = (V, E')$, являющийся деревом. Этот подграф F называют *остовом графа* G или *остовным деревом*. В случае если изначальный граф имеет n компонент связности, остовом называют подграф, который состоит из остовных деревьев для каждой компоненты связности.

Рассмотрим взвешенный граф, то есть граф $G = (V, E)$, ребрам которого приписаны некоторые числа $f(e)$ (задана функция f: $E \to \mathbb{R}$).

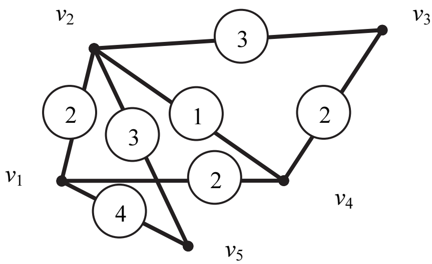


Рис. 27

Рассмотрим неориентированный взвешенный связный граф, состоящий из n вершин и m ребер. Пусть нам необходимо построить остов этого графа, который имеет наименьший суммарный вес ребер. Эту задачу называют *задачей о соединении городов* или *задачей о построении кратчайшего остова*. При наличии графа возможных дорог с указанием их стоимости нужно соединить все города, затратив наименьшие средства.

Для решения этой задачи используется *алгоритм Краскала* — на каждом шаге осуществляется выбор ребра минимального веса таким образом, чтобы не возникало циклов. Алгоритм заканчивает работу, когда выбрано $(n - 1)$-е ребро. Приведем формальное описание алгоритма.

Дан граф $G = (V, E)$.

1. Упорядочить ребра по возрастанию весов: e_1, e_2 ,…, e_m, $f(e_1) \leq \ldots \leq f(e_m)$.

2. Выбрать первое ребро, занести его в класс K_1. Классы означают компоненты связности построенной части дерева.

3. Выбрать следующее ребро $e = \{u, v\}$. Возможны четыре случая:

а) ни u, ни v не инцидентны никаким ребрам из имеющихся классов $K_1, \ldots, K_i$. Создается новый класс K_{i+1};

б) одна вершина v не инцидентна никакому ребру из классов $K_1, \ldots, K_i$, а u инцидентно некоторому ребру из K_j. Ребро $\{u, v\}$ добавляется в класс K_j;

в) вершины инцидентны ребрам из разных классов K_j и K_l. Классы объединяются в один класс, содержащий также ребро $\{u, v\}$;

г) обе вершины инцидентны ребрам из одного класса. Ребро не добавляется в дерево (иначе образуется цикл).

Шаг 3 повторяют $n - 2$ раза.

Пример. Упорядочим ребра графа, показанного на рис. 28, по возрастанию весов: $\{v_2, v_1\}$, $\{v_3, v_4\}$, $\{v_2, v_4\}$, $\{v_1, v_4\}$, $\{v_2, v_5\}$, $\{v_2, v_3\}$, $\{v_1, v_5\}$. Имеем $K_1 = \{\{v_2,\ v_1\}\}$.

Ребро $\{v_3,\ v_4\}$ — другая компонента связности: $K_2 = \{\{v_3,\ v_4\}\}$.

Вершины $\{v_2,\ v_4\}$ инцидентны ребрам из разных классов. Вместо классов K_1 и K_2 вводим $K_3 = \{\{v_2,\ v_4\}, \{v_2,\ v_1\}, \{v_3,\ v_4\}\}$.

Ребро $\{v_1,\ v_4\}$ образует цикл.

Добавляем $\{v_2,\ v_5\}$ в класс K_3.

Четыре выделенных ребра образуют минимальное остовное дерево (см. рис. 28).

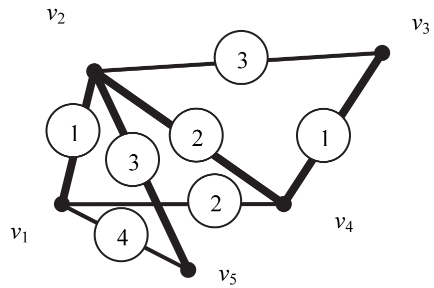


Рис. 28

5. ФУНДАМЕНТАЛЬНАЯ СИСТЕМА ЦИКЛОВ

Объединением графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называют объединение множеств вершин и ребер $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.

Цикл в графе — подграф, изоморфный C_n. Цикл обозначают как последовательность входящих в него вершин: $c = (v_1, v_2, ..., v_k)$, где $\{v_i, v_{i+1}\} \in E$, $i = 1, ..., k-1$, $\{v_k, v_1\} \in E$.

Полицикл в графе — объединение нескольких циклов графа, не имеющих общих ребер. Например, в графе, показанном на рис. 28, $(v_1, v_2, v_5) \cup (v_2, v_3, v_4)$ — полицикл.

Введем операцию сложения по модулю два ($\oplus$) для циклов фиксированного графа. Подграф $c^1 \oplus c^2$ — объединение циклов c^1 и c^2, но их общие ребра удаляются из объединения.

Например, $(v_1, v_2, v_4) \oplus (v_2, v_3, v_4) = (v_1, v_2, v_3, v_4)$, общее ребро $\{v_2, v_4\}$ удалено.

Множество полициклов P в любом графе с операцией сложения по модулю два ($\oplus$) и умножением на 0 и 1 является линейным пространством.

Фундаментальная система циклов (ФСЦ) — базис этого пространства P, то есть такая минимальная совокупность циклов, что любой цикл графа можно представить в виде суммы нескольких циклов фундаментальной системы.

Для нахождения ФСЦ, ассоциированной с заданным остовом графа, к остову добавляют одно ребро исходного графа, которое не входит в остов. При этом образуется ровно один цикл. Проделав эту процедуру со всеми не входящими в остов ребрами графа, получают систему циклов. Эта система фундаментальна, так как через циклы этой системы при помощи операции сложения по модулю два можно выразить любой цикл графа. Для того чтобы выразить любой цикл графа, нужно сложить циклы фундаментальной системы, соответствующие тем ребрам цикла, которые не входят в выбранный остов.

Пример. Рассмотрим граф из предыдущего примера (см. рис. 28) и используем уже построенное остовное дерево с ребрами $\{v_1, v_5\}$, $\{v_2, v_3\}$, $\{v_1, v_4\}$.

При добавлении ребра $\{v_1, v_5\}$ к остову возникает цикл $c^1 = (v_1, v_5, v_2)$, при добавлении $\{v_2, v_3\}$ — цикл $c^2 = (v_2, v_3, v_4)$, при добавлении $\{v_1, v_4\}$ — цикл $c^3 = (v_1, v_4, v_2)$.

Фундаментальная система состоит из циклов c^1, c^2, c^3, изображенных на рис. 29. Цикл $(v_1, v_2, v_3, v_4) = c^1 \oplus c^3$.

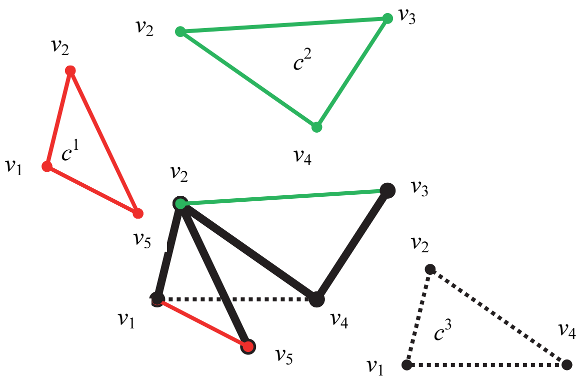


Рис. 29

Разрезом в графе называют минимальное по включению множество ребер, при удалении которого граф становится несвязным.

На множестве разрезов можно ввести ту же операцию $\oplus$, что и для циклов. *Фундаментальная система разрезов* (ФСР) — минимальная по включению система разрезов, через которую можно выразить любой разрез графа.

Для нахождения ФСР, ассоциированной с заданным остовом графа, из остова убирают одно ребро. При этом в остове образуются две новые компоненты связности. Множество ребер графа, соединяющих эти компоненты остова, вместе с удаленным ребром остова образуют разрез. Проделав эту процедуру со всеми ребрами, входящими в остов, получают систему разрезов. Она фундаментальна, так как через разрезы этой системы можно выразить любой разрез в графе.

Для того чтобы выразить произвольный разрез через фундаментальную систему, нужно сложить разрезы, соответствующие остовным ребрам данного разреза.

Пример. Рассмотрим тот же граф (см. рис. 28). ФСР будет состоять из следующих разрезов: $\{\{v_1, v_2\}, \{v_1, v_4\}, \{v_1, v_5\}\}$, $\{\{v_2, v_5\}, \{v_1, v_5\}\}$, $\{\{v_2, v_4\}, \{v_1, v_4\}, \{v_2, v_3\}\}$, $\{\{v_3, v_4\}, \{v_2, v_3\}\}$ — рис. 30.

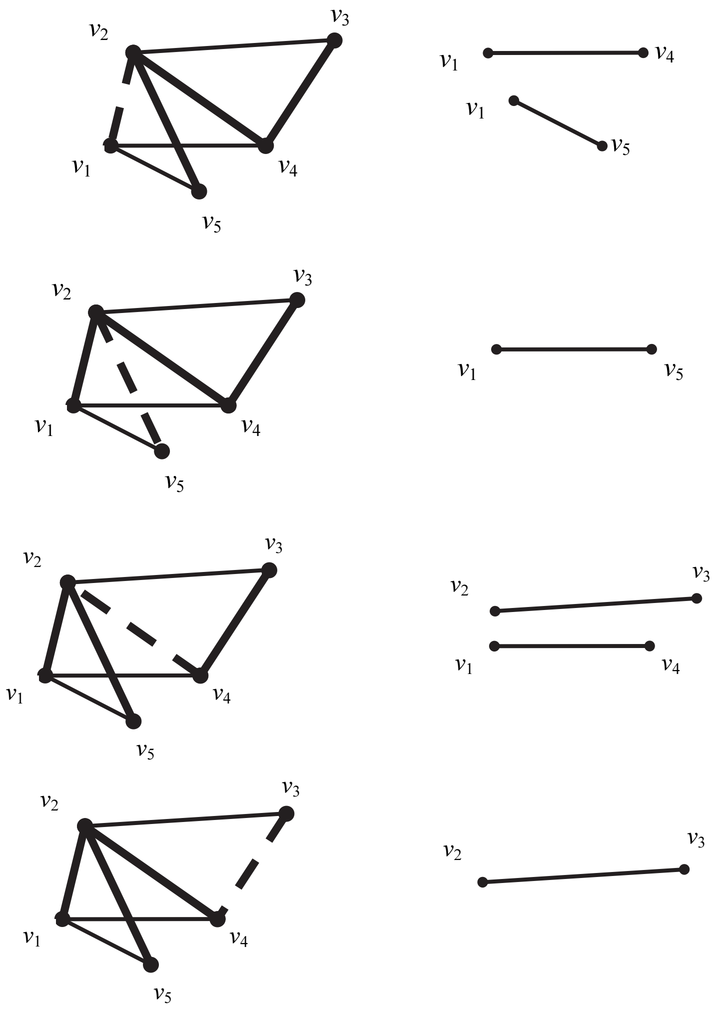


Рис. 30

6. ПОСТРОЕНИЕ ДЕРЕВА КРАТЧАЙШИХ ПУТЕЙ

Рассмотрим взвешенный граф. *Длиной пути* $\{v_1, v_2\}$, $\{v_2, v_3\}$, … …, $\{v_{k-1}, v_k\}$ называется суммарный вес ребер, входящих в путь, т. е. $f(v_1, v_2) + f(v_2, v_3) + \ldots + f(v_{k-1}, v_k)$.

Можно считать, что в невзвешенном графе все ребра имеют одинаковый вес — единицу. Тогда длиной пути будет число ребер в нем.

Расстояние $\rho(u,v)$ *между вершинами* u и v — длина кратчайшего пути, соединяющего эти вершины. Аналогично определяется расстояние для ориентированного графа.

Часто возникает задача выбрать из многих вариантов путей от одного пункта до других самый короткий, самый быстрый или самый дешевый (веса ребер интерпретируются как длина, время или стоимость). Эту задачу решают построением дерева кратчайших путей.

В графе выделена вершина — *корень*. *Дерево кратчайших путей* — остовное дерево графа, такое, что кратчайший путь в нем от любой вершины до корня имеет длину, равную расстоянию между этими вершинами в графе.

Рассмотрим взвешенный связный граф $G = (V, E)$, веса ребер обозначается $f(v_i, v_j)$. Требуется из данной вершины (корневой) построить кратчайшие пути до всех других вершин графа.

Алгоритм Дейкстры работает на графах со всеми неотрицательными весами, как на неориентированных, так и на ориентированных (в последнем случае учитывается направление). В ходе алгоритма вершинам будут приписываться метки μ, обозначающие длину кратчайшего пути. Опишем порядок действия алгоритма.

Корневой вершине приписывают метку $\mu_1 = 0$, остальным вершинам — метку бесконечность.

Шаг. Выбирают вершину с минимальной меткой (пусть это будет вершина v_i с меткой μ_i), ее метку называют постоянной. Перебирают все смежные с v_i вершины (выходящие из нее), метка которых не является постоянной. Допустим, вершина v_j смежна с v_i (или $(v_i, v_j) \in E$) и имеет метку μ_j. Если $\mu_i + f(v_i, v_j) < \mu_j$, то метку вершины v_j изменяют на $\mu_i + f(v_i, v_j)$, иначе оставляют ей старую метку μ_j.

Алгоритм заканчивает свою работу, когда все вершины имеют постоянные метки.

Далее из графа выбирают ребра искомого дерева. Для этого для каждой вершины определяют одну предшествующую ей (вершина v_i *предшествует* v_j, если $\mu_i + f(v_i, v_j) = \mu_j$). Каждую вершину соединяют с предшествующей, в результате чего образуется дерево кратчайших путей.

Пример реализации алгоритма Дейкстры показан на рис. 31. Постоянные метки обозначают числами со знаком +.

Алгоритм Форда (волновой алгоритм) работает как на графах с положительными весами, так и на графах с отрицательными весами при условии отсутствия цикла с отрицательным суммарным весом.

Алгоритм

Дан граф $G = (V, E)$, заданы веса ребер $f(v_i, v_j)$. Каждой вершине приписана метка бесконечность.

На предварительном шаге первой вершине присваивается $\mu_1 = 0$.

Шаг. Для каждой вершины v_i, метка которой μ_i была изменена на предыдущем шаге, выполняют следующие действия. Рассматривают все вершины v_j, смежные с v_i (выходят из нее). Если $\mu_i +$ $+ f(v_i, v_j) < \mu_j$, значение метки μ_j заменяют на $\mu_i + f(v_i, v_j)$.

Шаг алгоритма повторяют $|V| - 1$ раз.

Далее по меткам выбирают ребра, составляющие дерево кратчайших путей таким же образом, как и в алгоритме Дейкстры.

В результате работы алгоритма получают дерево кратчайших путей от выбранной вершины до всех остальных вершин связного графа и расстояние от нее до каждой другой вершины.

Ниже приведен пример применения алгоритма Форда к ориентированному графу.

Пример применения алгоритма Форда к ориентированному графу приведен на рис. 32.

Если этот алгоритм или алгоритм Дейкстры последовательно применить к каждой вершине графа, то можно получить *матрицу расстояний между вершинами.*

Центром связного графа называется такая его вершина, для которой наибольшее из расстояний до всех остальных вершин минимально. Заметим, что в графе возможно наличие нескольких центров, тогда указанное расстояние для всех них одинаково.

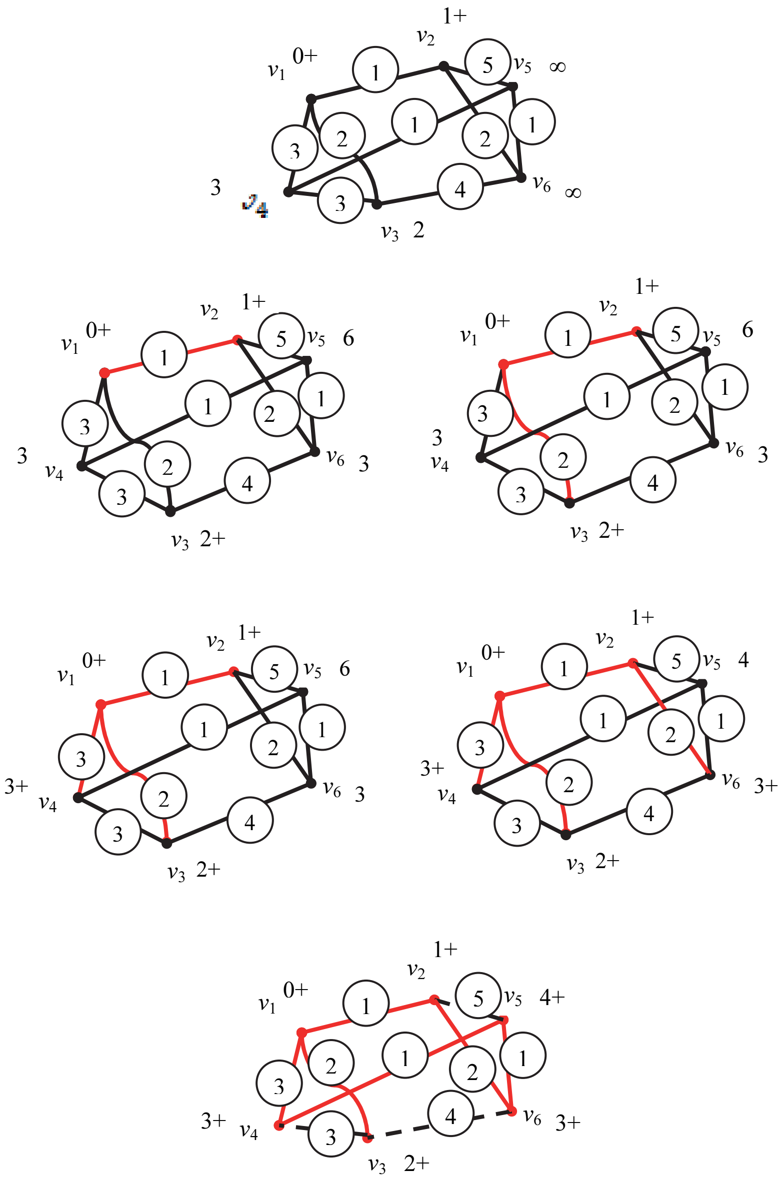
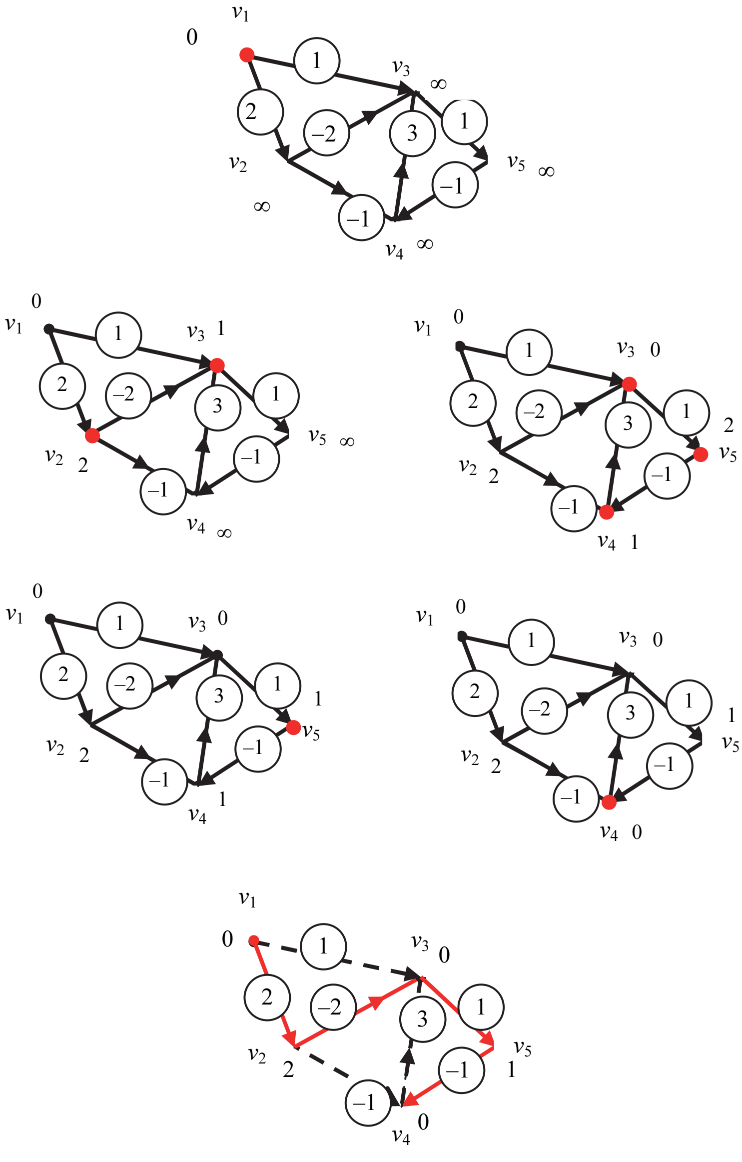

v_1 0+
v_2 1+
v_5 ∞
v_6 ∞
v_3 2
3 v_4
v_1 0+
v_2 1+
v_5 6
v_6 3
v_3 2+
3 v_4
v_1 0+
v_2 1+
v_5 6
v_6 3
v_3 2+
3 v_4
v_1 0+
v_2 1+
v_5 6
v_6 3
v_3 2+
3+ v_4
v_1 0+
v_2 1+
v_5 4
v_6 3+
v_3 2+
3+ v_4
v_1 0+
v_2 1+
v_5 4+
v_6 3+
v_3 2+
3+ v_4

Рис. 31



v_1 0
v_2 ∞
v_3 ∞
v_4 ∞
v_5 ∞
1
2
−2
3
1
−1
−1
v_1 0
v_2 2
v_3 1
v_4 ∞
v_5 ∞
v_1 0
v_2 2
v_3 0
v_4 1
v_5 2
v_1 0
v_2 2
v_3 0
v_4 1
v_5 1
v_1 0
v_2 2
v_3 0
v_4 0
v_5 1
v_1 0
v_2 2
v_3 0
v_4 0
v_5 1

Рис. 32

Радиус графа связан с центром, он равен максимальному из расстояний от каждой вершины до центра, rad = $\min_v \max_u \rho(u, v)$.

Диаметром связного графа называется максимальное расстояние между двумя его вершинами, diam = $\max_{u,v} \rho(u, v)$. Для определения центра, радиуса и диаметра графа нужно анализировать матрицу расстояний между вершинами.

Пример. Для графа, изображенного на рис. 33, найдем радиус и диаметр.

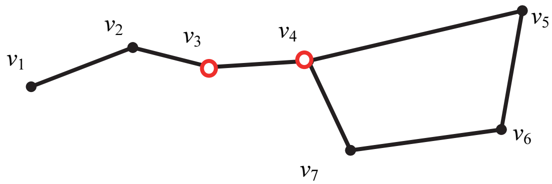


Рис. 33

Веса всех ребер равны единице.

Матрица расстояний для графа имеет следующий вид:

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	Максимальное значение
v_1	0	1	2	3	4	5	4	5
v_2	1	0	1	2	3	4	3	4
v_3	2	1	0	1	2	3	2	3
v_4	3	2	1	0	1	2	1	3
v_5	4	3	2	1	0	1	2	4
v_6	5	4	3	2	1	0	1	5
v_7	4	3	2	1	2	1	0	4

Диаметр графа равен 5 (расстояние между вершинами v_1 и v_6), радиус равен 3, в графе два центра — вершины v_3 и v_4.

7. КОНДЕНСАЦИЯ И БАЗА. ПОТОКИ В СЕТЯХ

Множество вершин ориентированного графа называют его *базой*, если из вершин этого множества достижимы все остальные вершины, и оно минимально по включению.

База существует в любом ориентированном графе. Любые две вершины базы не достижимы одна из другой. Вершины с нулевыми степенями захода обязаны принадлежать базе.

Для построения базы используется *конденсация* графа, то есть такой граф G^c, в котором каждая сильная компонента исходного графа стянута в одну вершину.

Конденсацию и базу графа находят следующим образом.

Построение графа конденсации.

1. Определяют для каждой вершины v множество достижимых из нее вершин $V_{v\,\text{д}}$ (это можно сделать, например, с помощью алгоритмов поиска в глубину или в ширину).

2. Выделяют наибольшие по числу включений классы вершин V_1, V_2, ..., V_k, достижимых друг из друга: $V = V_1 \cup V_2 \cup \ldots \cup V_k$. Эти классы будут вершинами графа G^c.

Вершины V_1 и V_2 графа G^c соединяют дугой, если в исходном графе существовала дуга (v, u), где $v \in V_1$, $u \in V_2$.

Множество вершин, содержащих ровно одну вершину из множества новых вершин V_1, V_2, …, V_k, имеющих нулевую полустепень захода, образует базу.

В качестве примера на рис. 34 показан исходный граф, а на рис. 35 — его конденсация.

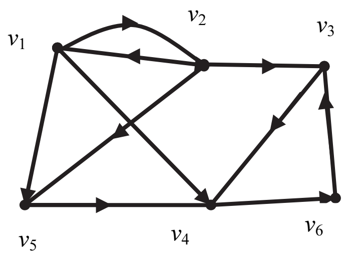


Рис. 34

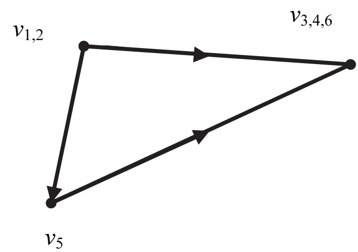


Рис. 35

Множества достижимых вершин:

$V_{д1} = \{v_1, v_2, v_3, v_4, v_6, v_5\}$; $V_{д2} = \{v_2, v_1, v_4, v_4, v_6, v_5\}$;

$V_{д3} = \{v_3, v_4, v_6\}$; $V_{д4} = \{v_4, v_6, v_3\}$;

$V_{д5} = \{v_5, v_4, v_6, v_3\}$; $V_{д6} = \{v_6, v_3, v_4\}$.

Классы взаимно достижимых вершин:

$V_1 = \{v_1, v_2\}$; $V_2 = \{v_3, v_4, v_6\}$; $V_3 = \{v_5\}$.

База состоит, например, из вершины v_1.

Сетью называют связный ориентированный граф, в котором выделены непересекающиеся множества: вершин с нулевой степенью захода (истоки) и с нулевой степенью исхода (стоки).

Взвешенная сеть — сеть, в которой каждому ребру приписано некоторое положительное число («пропускная способность ребра») $f(u, v)$, считаем, что вес отсутствующих ребер равен нулю.

Некоторым дугам сети $\{u, v\}$ припишем числа $p_{uv} \geq 0$, называемые потоками, удовлетворяющие следующим условиям:

• поток не превышает пропускную способность: $p_{uv} \leq f(u, v)$;

• для каждой вершины суммарный поток дуг, входящих в вершину, равен сумме потоков дуг, выходящих из нее: $\sum\limits_u p_{uv} = \sum\limits_w p_{vw}$;

• суммарный поток дуг, выходящих из всех истоков, равен суммарному потоку дуг, входящих в стоки: $\sum\limits_v p_{Iv} = \sum\limits_w p_{wS}$.

Величину $P = \sum\limits_v p_{Iv}$ называют *потоком в сети*. Наибольшее значение P называют *максимальным потоком в сети.*

Задача определения максимального потока в сети возникает, например, при проектировании дороги с учетом напряженности транспортных потоков, водопроводной либо канализационной сети и т. п. Будем решать эту задачу с помощью **алгоритма Форда — Фалкерсона**:

1) полагают поток равным нулю: $P = 0$;

2) находят путь из источника в сток. Если таких путей нет, то алгоритм заканчивает свою работу. Среди всех входящих в найденный путь дуг находят дугу с минимальным весом (с минимальной пропускной способностью), пусть этот вес равен a. Вычи-

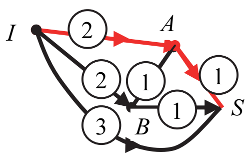
тают a из весов всех входящих в найденный путь дуг и прибавляют его к текущему значению пропускной способности: $P = P + a$.

Также прибавляют вес a к пропускным способностям дуг, имеющих обратные направления по отношению к дугам, входящим в рассмотренный путь (для каждой рассмотренной дуги (u, v) обратной будет (v, u)). Если такой дуги не было, считают, что дуга имела пропускную способность 0, и вновь проводят ее с пропускной способностью a.

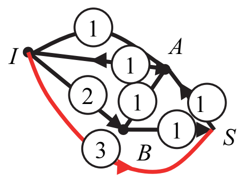
Дуги с нулевым весом можно не изображать.

Повторяют действия, описанные в п. 2, до тех пор, пока не будут перебраны все возможные пути и сеть не станет непроходимой. Получившееся значение P и есть максимальный поток.

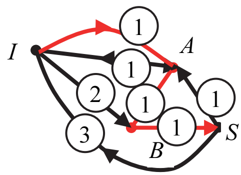
Пример использования алгоритма показан на рис. 36.



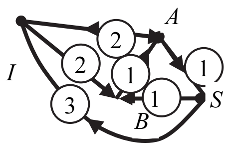
Путь $I \to A \to S$
$a = 1$, $P = 1$



Путь $I \to S$
$a = 3$, $P = 4$



Путь $I \to A \to B \to S$
$A = 1$, $P = 5$



Путей из I в S нет,
$P_{max} = 5$

Рис. 36

Разрез для сети — минимальное по включению множество дуг, при удалении которых сеть становится непроходимой из источников в стоки. *Минимальный разрез* — разрез с минимальным суммарным весом.

Теорема Менгера для сетей. Пропускная способность сети равна величине минимального разреза.

Для данного примера минимальным разрезом является $\{(A, S), (B, S), (I, S)\}$. Его суммарный вес равен 5.

8. ПЛАНАРНОСТЬ. ВЕРШИННАЯ РАСКРАСКА

Планарность. Иногда возникает потребность изобразить граф на плоскости без пересечения ребер. Если это удается, то полученный геометрический граф называют *плоским*. Любой граф, изоморфный плоскому, называется *планарным*.

Задача о трех колодцах. Это древняя восточная задача. Есть три дома и три колодца. Жители каждого дома хотят брать воду из всех трех колодцев, но все они поссорились и не хотят встречаться. Как проложить тропинки от домов к колодцам?

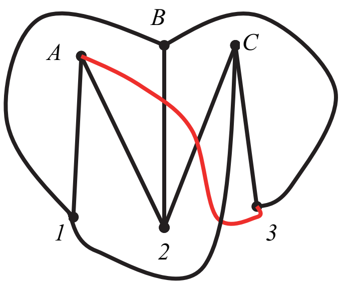


Рис. 37

Эта задача не имеет решения — граф $K_{3,3}$, который соответствует этой задаче, непланарен (рис. 37). Непланарным является также граф K_5.

Теорема Понтрягина — Куратовского. Граф является планарным тогда и только тогда, когда он не содержит подграфов, изоморфных $K_{3,3}$ и K_5.

Задача правильной раскраски графа. Пусть $G = (V, E)$ — произвольный граф, k — натуральное число. Произвольная функция вида $f\colon V \to \{1, 2, 3, \ldots, k\}$ называется вершинной ***k*-раскраской** графа G. Раскраска называется **правильной**, если $f(u) \neq f(v)$ для любых смежных вершин u и v.

Очевидно, что для правильной раскраски полного графа, содержащего n вершин, требуется n красок. Для правильной раскраски любого дерева требуется две краски, для C_n требуется две краски, если n четно, и три, если n нечетно.

Хроматическое число графа χ_G — минимальное число красок, необходимое для правильной раскраски графа.

Задача о проектировании коробки скоростей. Коробка скоростей — механизм для изменения частоты вращения ведомого вала при постоянной частоте вращения ведущего. Это изменение достигается в результате того, что находящиеся внутри коробки шестерни (зубчатые колеса) вводятся в зацепление специальным образом. Задача состоит в минимизации числа валов, на которых размещаются шестерни.

Построим граф, в котором вершины взаимно однозначно сопоставлены с шестернями. Ребра проводим между теми вершинами, для которых соответствующие шестерни по каким-либо причинам не могут быть насажены на один вал. Минимальная правильная раскраска такого графа дает минимальное число валов — вершины, окрашенные в один цвет, могут быть насажены на один вал.

Теорема Хивуда о пяти красках. Для любого планарного графа хроматическое число не превышает пяти.

Гипотеза Кэли. В 1879 г. британский математик А. Кэли опубликовал в Трудах Лондонского географического общества статью, посвященную проблеме раскраски карт, в которой была сформулирована гипотеза о четырех красках — всякая карта 4-раскрашиваема.

Заметим, что любую географическую (политическую) карту можно взаимно однозначно сопоставить с планарным графом, т. е. с графом, который может быть изображен на плоскости без пересечения ребер. Вершины в нем будут соответствовать странам, а ребра — проводиться между граничащими странами.

Алгоритм последовательной раскраски. Раскраску начинают с любой вершины, которую красят в первый цвет. Далее на каждом этапе выбирают для окраски вершину, смежную с какой-либо уже окрашенной, и красят ее в следующий цвет. Так продолжают до тех пор, пока все вершины не будут окрашены.

Этот алгоритм дает правильную раскраску, но не всегда минимальную. К сожалению, не существует эффективного алгоритма для получения минимальной правильной раскраски, в общем случае для этой задачи необходимо применять полный перебор вершин.

Пример правильной раскраски графа показан на рис. 38.

Хроматический многочлен графа $f(k)$ — функция числа красок k, равная числу способов раскраски графа в k цветов.

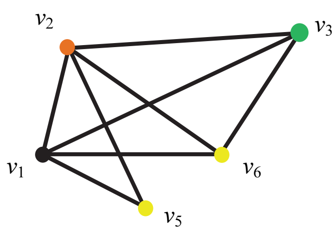


Рис. 38

Если $k < \chi_G$, то $f(k) = 0$.

Хроматическое число полного графа равно числу его вершин $n = |V|$. Хроматический многочлен полного графа для $k \geq n$ имеет вид $f(k) = \dfrac{k!}{(k-n)!}$.

Чтобы найти хроматический многочлен произвольного графа, нужно разложить его на полные графы.

Введем операцию *слияния* вершин u и v: вместо этих вершин в графе вводится вершина uv, с которой смежны те вершины, которые были смежны с u или с v.

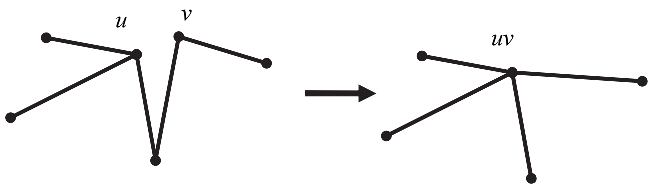


Рис. 39

Соединением вершин называют добавление в граф ребра $\{u, v\}$.

Лемма. Пусть граф G_1 получен из графа G соединением несмежных вершин u и v, а граф G_2 — слиянием вершин u и v. Тогда $f_G(k) = f_{G_1}(k) + f_{G_2}(k)$.

Утверждение леммы основывается на том, что в хроматическом многочлене учитываются все способы правильной раскраски, а каждая пара несмежных вершин может быть раскрашена либо в один и тот же цвет, либо в разные цвета. Таким образом можно найти хроматический многочлен для произвольного графа.

Пример. Рассмотрим граф G (рис. 40). Пользуясь леммой, получим

$$f_G = f_{G_1} + f_{G_2} = f_{G_{11}} + f_{G_{12}} + f_{G_{21}} + f_{G_{22}} =$$
$$= f_{G_{111}} + f_{G_{112}} + f_{G_{12}} + f_{G_{21}} + f_{G_{22}}.$$

Так как граф G_{111} — полный граф K_5, G_{22} есть K_3, а графы G_{112}, G_{11} и G_{21} — K_4, и для них хроматические многочлены известны, следовательно,

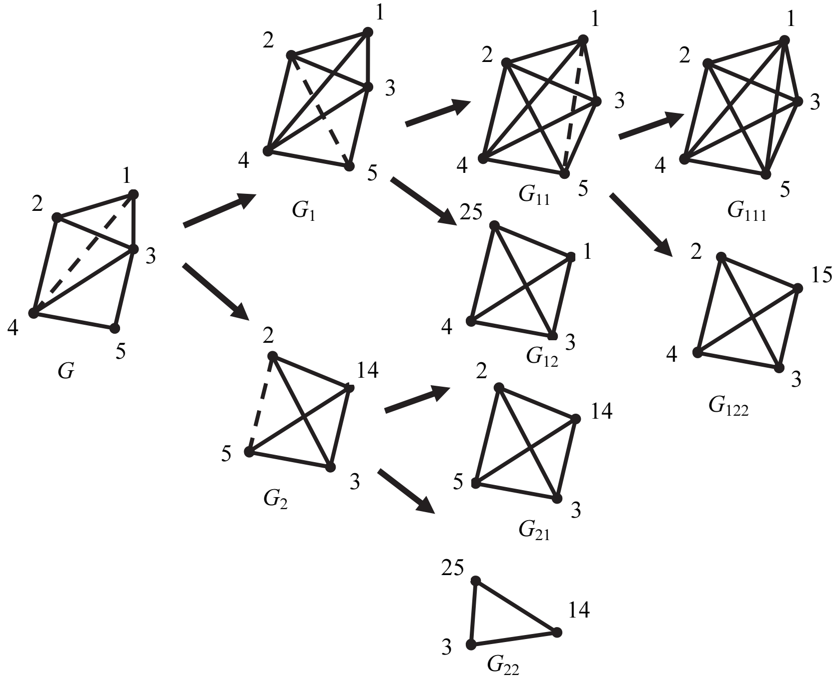


Рис. 40

$$f_G = k(k-1)(k-2)(k-3)(k-4) + 3k(k-1)(k-2)(k-3) +$$
$$+ k(k-1)(k-2) = k(k-1)(k-2)^3.$$

Заметим, что, зная хроматический многочлен, можно найти и хроматическое число: это минимальное число k, такое что $f_G(k) > 0$. В данном случае $\chi_G = 3$.

ЛИТЕРАТУРА

Белоусов А.И., *Ткачев С.Б.*, Дискретная математика. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. (Серия математика в техническом университете. Вып. XIX).

Лекции по теории графов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич. М.: Наука, 1990.

Оре О. Теория графов / пер. с англ. М.: Наука, 1968.

Исмагилов Р.С., *Калинкин А.В.*, *Станцо В.В*. Графы: учеб. пособие по курсу «Дискретная математика». М.: Изд-во МГТУ им. Н.Э. Баумана, 1999.

ОГЛАВЛЕНИЕ

Предисловие 3
1. Основные определения и понятия 4
2. Способы задания графа 11
3. Задачи об обходах графа 14
4. Дерево. Минимальное остовное дерево 20
5. Фундаментальная система циклов 23
6. Построение дерева кратчайших путей 26
7. Конденсация и база. Потоки в сетях 31
8. Планарность. Вершинная раскраска 34
Литература 38

Учебное издание

Бояринцева Татьяна Евгеньевна
Мастихина Анна Антоновна

Теория графов

Редактор *С.А. Серебрякова*
Корректор *А.К. Еникеева*
Компьютерная верстка *С.А. Серебряковой*

Подписано в печать 15.07.2014. Формат 60×84/16.
Усл. печ. л. 2,32. Тираж 500 экз. Изд. № 2. Заказ

Издательство МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.
press@bmstu.ru
www.baumanpress.ru

Отпечатано в типографии МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.
baumanprint@gmail.com