

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ**

**Г.Г. ШВАЧИЧ, Г.М. БАРТЕНЄВ,
О.В. ОНИЩЕНКО, В.В. ТОЛСТОЙ**

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

Частина III. Основи теорії графів

**Затверджено на засіданні Вченої ради академії
як навчальний посібник. Протокол № 1 від 27.01.2014**

Дніпропетровськ НМетАУ 2014

УДК 543. 211/.205+543.4

Г.Г. Швачич, Г.М. Бартенєв, О.В. Онищенко, В.В. Толстой. Основи дискретної математики. Частина III. Основи теорії графів: Навч. посібник (російською мовою). – Дніпропетровськ: НМетАУ, 2014. – 67 с.

Навчальний посібник органічно поєднує теоретичні положення дискретної математики з їх використанням для розв'язку практичних завдань відповідного напрямку.

Наведені приклади розв'язування типових задач і задач для самостійної роботи студентів з розділу дисципліни «Основи дискретної математики»: основи булевої алгебри.

Призначений для студентів напрямку 6.050101 – комп'ютерні науки.

Лл. 101. Табл. 9. Бібліогр.: 15 найм.

Друкується за авторською редакцією.

Відповідальний за випуск Г.Г. Швачич, д-р техн. наук, проф.

Рецензенти: А.В. Поліщук, канд. техн. наук, доц. (ДНУЖТ)
О.Г. Холод, канд. техн. наук, доц. (Державний Університет ім. А. Нобеля)

© Національна металургійна академія України, 2014

© Швачич Г.Г., Бартенєв Г.М.,
Онищенко О.В., Толстой В.В., 2014

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 4 |
| 1. ОПРЕДЕЛЕНИЕ ГРАФА..... | 6 |
| 2. КЛАССИФИКАЦИЯ ГРАФОВ..... | 8 |
| 3. ОТНОШЕНИЕ МЕЖДУ РЕБРАМИ И ВЕРШИНАМИ ГРАФА..... | 10 |
| 4. СТЕПЕНИ ВЕРШИН ГРАФА | 11 |
| 5. СПОСОБЫ ЗАДАНИЯ ГРАФОВ..... | 13 |
| 6. МАРШРУТ, ПУТЬ, ЦЕПЬ, ЦИКЛ | 16 |
| 7. СВЯЗНОСТЬ ГРАФОВ..... | 19 |
| 8. ПОДГРАФ. ЧАСТИ ГРАФА | 20 |
| 9. ДЕРЕВО И ЛЕС..... | 21 |
| 10. ИЗОМОРФИЗМ ГРАФОВ | 24 |
| 11. ПЛАНАРНОСТЬ ГРАФОВ | 26 |
| 12. ДВУДОЛЬНЫЕ ГРАФЫ | 28 |
| 13. ПАРОСОЧЕТАНИЯ В ГРАФЕ | 29 |
| 14. БИНАРНЫЕ ОТНОШЕНИЯ И ГРАФЫ..... | 32 |
| 15. ЭЙЛЕРОВЫ ГРАФЫ..... | 37 |
| 16. ГАМИЛЬТОНОВЫ ГРАФЫ | 41 |
| 17. НЕКОТОРЫЕ ЗАДАЧИ НА ГАМИЛЬТОНОВЫХ ГРАФАХ | 44 |
| 18. ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ НА ГРАФАХ | 47 |
| 18. 1. ПОСТРОЕНИЕ СЕТИ НАИМЕНЬШЕЙ ДЛИНЫ. АЛГОРИТМ КРАСКАЛА | 48 |
| 18.2. ПОИСК КРАТЧАЙШИХ ПУТЕЙ ИЗ ДАННОЙ ВЕРШИНЫ ГРАФА КО ВСЕМ ОСТАЛЬНЫМ | 53 |
| 18.3. НАХОЖДЕНИЕ МАКСИМАЛЬНОГО ПОТОКА В СЕТИ | 60 |
| ЛИТЕРАТУРА..... | 66 |

ВВЕДЕНИЕ

Теория графов представляет собой раздел математики, имеющий широкое практическое применение. В её терминах формулируется большое число задач, связанных с дискретными объектами. Такие задачи возникают при проектировании интегральных схем и схем управления, электрических цепей, блок-схем программ, в экономике, статистике, химии, биологии и в других областях. Теория графов становится одной из существенных частей математического аппарата кибернетики, языком дискретной математики.

В отличие от других научных дисциплин, теория графов имеет вполне определенную дату рождения. Первая работа по теории графов, написанная швейцарским, немецким и российским математиком Леонардом Эйлером (1707-1783), была опубликована в 1736 году в Трудах Академии наук в Санкт-Петербурге. Исследование Эйлера было проведено в связи с популярной в то время задачей о кенигсбергских мостах (рис. 1). Эта головоломка была очень популярна у жителей Кенигсберга и многие пытались решить её эмпирически, снуя туда и обратно по городским мостам. Об этом Л. Эйлер пишет в письме итальянскому математику и инженеру Маринони:

«Некогда мне была предложена задача об острове, расположенном в городе Кенигсберге и окруженном рекой, через которую перекинута семь мостов. Спрашивается, может ли кто-нибудь непрерывно обойти их, проходя только однажды через каждый мост. И тут же мне было сообщено, что никто еще до сих пор не мог это проделать, но никто и не доказал, что это невозможно.

Вопрос этот, хотя и банальный, показался мне, однако, достойным внимания тем, что для его решения недостаточны ни геометрия, ни алгебра, ни комбинаторное искусство. После долгих размышлений я нашел легкое правило, основанное на вполне убедительном доказательстве, с помощью которого можно во всех задачах такого рода тотчас же определить, может ли быть совершен такой обход через какое угодно число и как угодно расположенных мостов или не может. Кенигсбергские же мосты расположены так, что их можно представить на следующем рисунке, на котором A обозначает остров, а B , C и D – части континента, отделенные друг от друга рукавами реки. Семь мостов обозначены буквами a , b , c , d , e , f , g »(рис. 1).

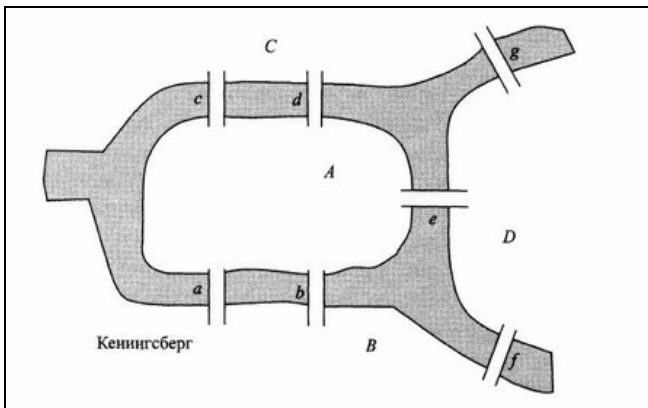


Рис. 1

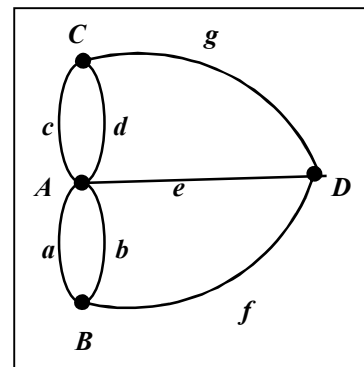


Рис. 2

Поскольку в предложенной задаче существенными элементами являются лишь переходы через мосты, план задачи можно свести к простой схеме (рис. 2). Здесь частям суши A, B, C, D соответствуют точки с тем же названием (вершины), а мостам a, b, c, d, e, f – одноименные линии (рёбра).

Эйлер, которому в ту пору шёл двадцать первый год, показал, что такого маршрута не существует, то есть в данной постановке задача неразрешима. Этот результат более ста лет был единственным полученным результатом в теории графов.

Интерес к проблемам теории графов возродился около середины XIX–го столетия и был сосредоточен, главным образом, в Англии. Имелось много причин для такого оживления изучения графов. Естественные науки оказали свое влияние на это, благодаря исследованиям электрических сетей, моделей кристаллов и структур молекул. Развитие формальной логики привело к изучению бинарных отношений в форме графов. Большое число популярных головоломок поддавалось формулировкам непосредственно в терминах графов, и это приводило к пониманию, что многие задачи такого рода содержат некоторое математическое ядро, важность которого выходит за рамки конкретного вопроса. Наиболее знаменитая среди этих задач – *проблема четырех красок*, впервые поставленная перед математиками де Морганом в 1852 году. Суть задачи такова: можно ли любую карту на плоскости или на поверхности шара раскрасить только четырьмя разными красками так, чтобы никакие две смежные страны (которые имеют общую границу, отличную от точки) не были бы одного и того же цвета?

Никакая другая проблема не вызывала столь многочисленных и остроумных работ в области теории графов. Благодаря своей простой формулировке и

раздражающей неуловимости она до сих пор не решена и остается мощным стимулом исследований различных свойств графов.

XX-е столетие было свидетелем неуклонного развития теории графов, которая за последние десять – двадцать лет вступила в новый период интенсивных разработок. В этом процессе явно заметно влияние запросов новых областей приложений: теории игр и программирования, теории передачи сообщений, электрических сетей и контактных цепей, а также проблем биологии и психологии.

Термин *граф* (от латинского слова *графіо* – писать) впервые появился в книге выдающегося венгерского математика Д. Кёнига в 1936 году.

В заключении следует отметить, что *терминология* теории графов и поныне не определена строго. В частности, нет единого мнения о том, какой из двух терминов *граф* или *сеть* следует применять. Аналогичная ситуация с термином *вершина*, *точка* или *полюс*. И таких примеров немало.

1. ОПРЕДЕЛЕНИЕ ГРАФА

Пример. Встретились пять друзей и обменялись рукопожатиями. Сколько всего рукопожатий было сделано?

Поставим каждому из пяти друзей в соответствие точку плоскости с номерами от 1 до 5. Тогда рукопожатию будет соответствовать отрезок или часть кривой, соединяющей конкретные точки. Количество таких отрезков и будет равно числу рукопожатий. Их будет 10 (рис. 1.1).

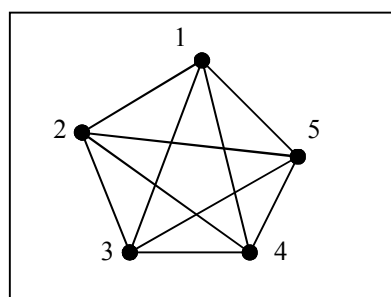


Рис. 1.1

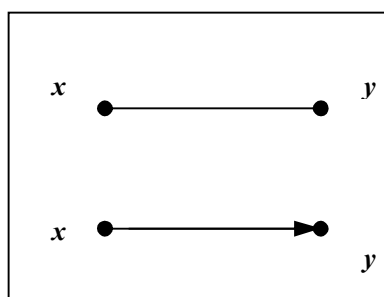


Рис. 1.2

Из приведенного примера очевидно, что мы имеем дело с двумя видами объектов – точками и линиями, их соединяющими. Эти же объекты изучает также и геометрия. Однако, если там важны длины линий, их взаимное расположение и т.д., то здесь это не имеет значения. Важно лишь наличие точек и связей между ними.

Рассмотрим два множества, которые обозначим V и E .

Множество V – множество некоторых объектов, которые можно изобразить точкой (например, людей, населённые пункты и т.д.). Это множество назовём множеством *вершин* (*точек* или *узлов*).

Множество E – множество линий, которые связывают вершины. Назовём его множеством *рёбер*.

Формально *граф* определяется как совокупность двух множеств – множества вершин и множества рёбер, соединяющих эти вершины: $G = \langle V, E \rangle$. При этом множество V не может быть пустым, т.е. $V \neq \emptyset$, а $E \subset V \times V$ (оно может быть пустым).

Количество вершин графа $n = |V|$ называют порядком графа, количество рёбер обозначают $m = |E|$.

Графы обозначают буквами G, H, Γ и другими и изображают с помощью диаграмм, где вершины изображены точками или кружочками, а рёбра – отрезками прямой или другими линиями. Все вершины графа должны быть идентифицированы, например $V = \{v_1, v_2, \dots, v_n\}$. На рис. 1.2 показаны два ребра: верхнее не имеет направления, обозначается парой $\{x, y\}$, где x и y – вершины; нижнее направлено от вершины x к y , обозначается $\langle x, y \rangle$.

Иногда требуется идентифицировать все рёбра графа. Тогда граф определяется как совокупность трёх множеств: $G = \langle V, E, O \rangle$, где $V = \{v_1, v_2, \dots, v_n\}$ – множество вершин; $E = \{e_1, e_2, \dots, e_m\}$ – множество рёбер графа; $O = \{(v_i, v_j), i=1, \dots, n; i=1, \dots, m\}$ – множество упорядоченных или нет пар вершин, соответствующих рёбрам множества E .

Часто вместо скобок « $\{\}$ » и « $\langle \rangle$ » применяют обычные круглые скобки, оговаривая при этом, ориентированы рёбра графа или нет.

Задача 1.1. Изобразить следующие графы с неориентированными рёбрами:

- а) $G = \langle V, E \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{(v_1, v_2), (v_2, v_3), (v_5, v_4), (v_5, v_2), (v_5, v_1), (v_4, v_2)\}$;
- б) $G = \langle V, E, O \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, $O = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_4, v_2), (v_5, v_4), (v_4, v_3)\}$.

Решение. Схема графа а) представлена на рис. 1.3; графа б) – на рис. 1.4.

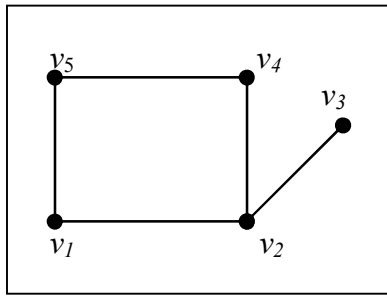


Рис. 1.3

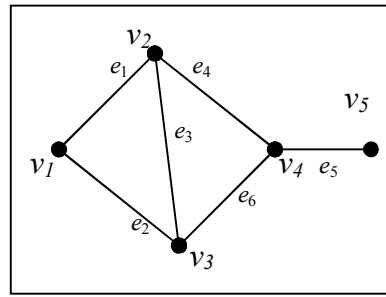


Рис. 1.4

2. КЛАССИФИКАЦИЯ ГРАФОВ

Существуют два основных признака, по которым можно классифицировать графы.

Первый признак – это наличие или отсутствие ориентации у рёбер. *Неориентированный граф* – это граф, у которого рёбра не имеют направления. Такие рёбра описываются неупорядоченной парой $\{x, y\}$. *Ориентированный граф* (или *орграф*) – это граф, рёбра которого имеют направление и могут быть описаны упорядоченной парой $\langle x, y \rangle$. Ориентированное ребро называют *дугой*.

Существуют графы, у которых наряду с ориентированными рёбрами (дугами) имеются и неориентированные. Они называются *смешанными*. При решении задач такие графы сводят к орграфам.

Второй признак – это наличие или отсутствие кратных рёбер у графа. *Кратными* (или *параллельными*) называются рёбра, которые связывают одни и те же вершины (рис. 2.1).

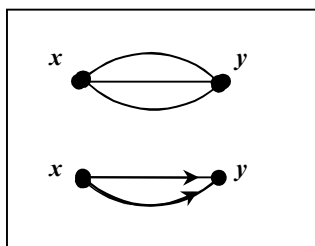


Рис.2.1

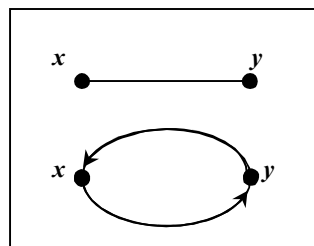


Рис.2.2

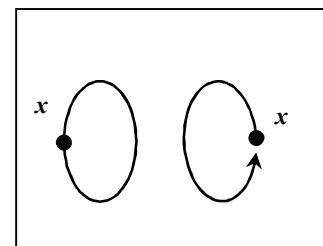


Рис.2.3

Всякое неориентированное ребро можно заменить парой рёбер противоположной направленности (рис. 2.2). Если ребро исходит из какой-либо вершины и входит в неё же, то такое ребро называется *петля* (рис. 2.3). Направление для петли, как правило, значения не имеет.

Мультиграф – граф, имеющий кратные рёбра. *Псевдограф* – граф, имеющий петли. *Простой граф* – граф, не имеющий кратных рёбер и петель.

Задача 2.1. Описать граф, изображенный на рис. 2.4.

Решение. $G = \langle V, E, O \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4, e_5\}$, $O = \{\langle v_2, v_1 \rangle, \langle v_1, v_2 \rangle, \langle v_2, v_3 \rangle, \langle v_1, v_3 \rangle, \langle v_2, v_4 \rangle\}$.

Задача 2.2. Определить вид графа и изобразить его схему:

$G = \langle V, E, O \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$, $O = \{\langle v_1, v_2 \rangle, \{v_2, v_2\}, \langle v_2, v_3 \rangle, \langle v_3, v_4 \rangle, \{v_5, v_4\}, \{v_2, v_4\}, \langle v_4, v_4 \rangle, \langle v_3, v_5 \rangle\}$.

Решение. Данный граф является смешанным и содержит петли (рис. 2.5).

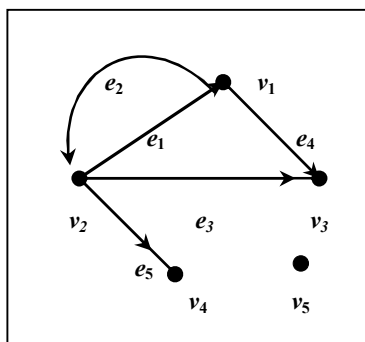


Рис. 2.4

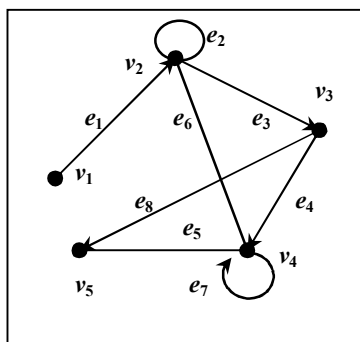


Рис. 2.5

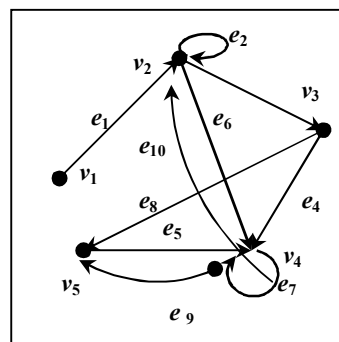


Рис. 2.6

Задача 2.3. Преобразовать граф предыдущей задачи к орграфу и изобразить его схему.

Решение. Для этого ребро $\{v_5, v_4\}$ заменим парой дуг $\langle v_5, v_4 \rangle$ с именем e_5 и $\langle v_4, v_5 \rangle$ с именем e_9 ; ребро $\{v_2, v_4\}$ – дугами $\langle v_2, v_4 \rangle$ (имя e_6) и $\langle v_4, v_2 \rangle$ (имя e_{10}). Петлю e_2 , не являющуюся ориентированной, можно считать ориентированной. (рис. 2.6).

Задачи для самостоятельного решения.

1. Изобразить схему графа: $G = \langle V, E \rangle$; $V = \{v_1, v_2, v_3, v_4\}$, $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_4, v_1\}, \{v_2, v_4\}, \{v_4, v_4\}\}$.
2. Изобразить схему графа: $G = \langle V, E \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $E = \{\langle v_1, v_2 \rangle, \langle v_2, v_5 \rangle, \langle v_5, v_5 \rangle, \langle v_1, v_5 \rangle, \langle v_3, v_4 \rangle, \{v_6, v_1\}, \langle v_6, v_6 \rangle, \langle v_4, v_2 \rangle, \langle v_4, v_4 \rangle, \langle v_3, v_3 \rangle\}$.
3. Изобразить схему графа: $G = \langle V, E, O \rangle$; $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$; $O = \{\langle v_1, v_3 \rangle, \langle v_2, v_3 \rangle, \langle v_3, v_3 \rangle, \langle v_3, v_4 \rangle, \langle v_2, v_4 \rangle, \langle v_4, v_2 \rangle, \langle v_5, v_4 \rangle, \langle v_4, v_5 \rangle, \langle v_2, v_5 \rangle, \langle v_1, v_5 \rangle\}$.

3. ОТНОШЕНИЕ МЕЖДУ РЕБРАМИ И ВЕРШИНАМИ ГРАФА

Всякое ребро *инцидентно* двум вершинам, которые оно соединяет. В свою очередь, вершины x и y инцидентны ребру e . Две вершины называются *смежными*, если они принадлежат одному и тому же ребру. На рис. 3.1 ребро e является инцидентным вершине x и вершине y , а сами вершины смежные.

Два ребра называются *смежными*, если они имеют общую вершину. На рис. 4.2 рёбра e_1 и e_2 смежные, поскольку вершина y для них общая.

Таким образом понятие *инцидентность* связывает объекты различных типов (вершина и ребро), а понятие *смежность* – объекты одного и того же типа (вершина и вершина, ребро и ребро).

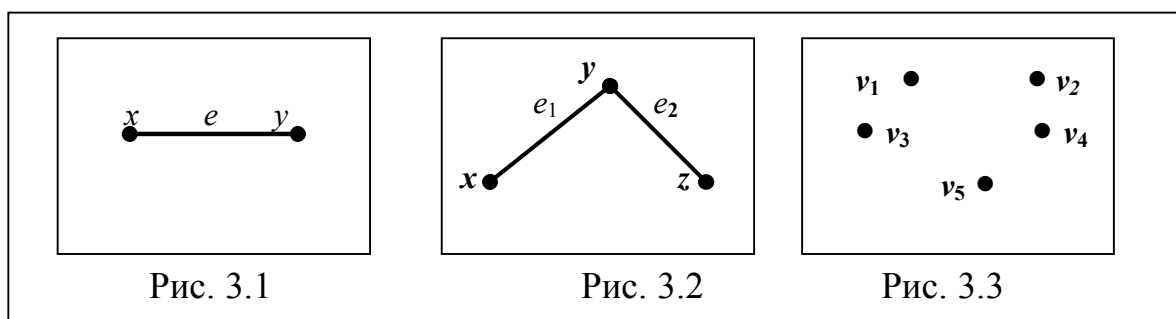


Рис. 3.1

Рис. 3.2

Рис. 3.3

Граф, не имеющий рёбер, называется *пустым* графом и обозначается O_n . Граф, у которого нет вершин, называется *нуль-графом*. На рис. 3.3 изображен пустой граф $O_5 = G\langle V, E \rangle$, у которого $V \neq \emptyset$, $E = \emptyset$.

Вершина графа, не инцидентная ни одному ребру, называется *изолированной*. Такая вершина не входит в множество пар E . На рис. 2.4 это вершина v_5 . Если вершина инцидентна только одному ребру, то она называется *висящей*. Такая вершина входит в множество пар E только один раз. На рис. 2.4 это вершина v_4 .

Граф называется *полным*, если любые его две вершины являются смежными, то есть любые две вершины такого графа соединены ребром. Полный граф порядка n (имеющий n вершин) может обозначаться символом K_n . Количество рёбер такого графа равно числу сочетаний из n элементов по 2:

$$m = C_n^2 = \frac{n(n-1)}{2}.$$

У полного ориентированного графа количество дуг будет в два раза больше, чем у неориентированного:

$$m = C_n^2 = n(n-1).$$

Задача 3.1. Построить полные графы K_n для $n = 2, 3, 4, 5$.

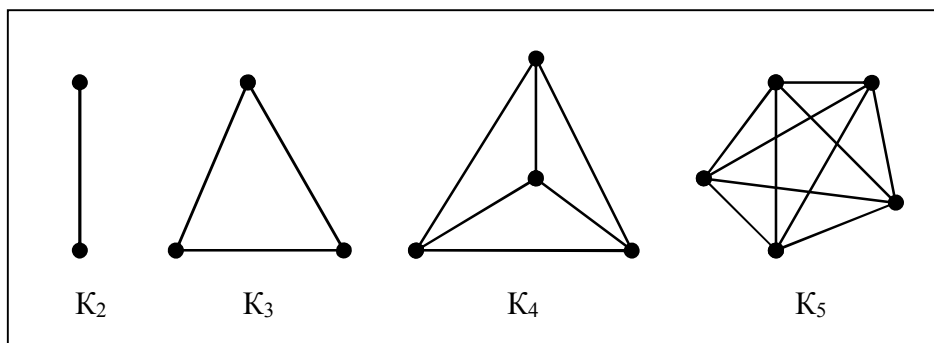


Рис. 3.4

Задача 3.2. Построить полные орграфы порядка 2 и 3.

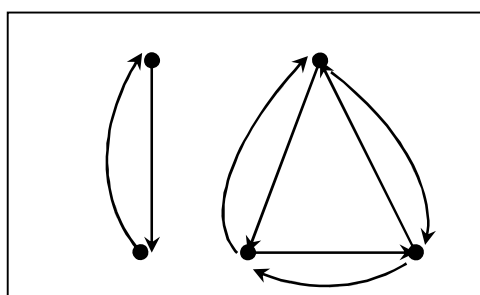


Рис. 3.5

Решение. Соответствующие орграфы изображены на рис. 3.5.

4. СТЕПЕНИ ВЕРШИН ГРАФА

Рассмотрим неориентированный граф $G = \langle V, E \rangle$ без петель. Пусть x – какая-либо из его вершин, $x \in V$.

*Степенью вершины x графа G называется число рёбер, инцидентных этой вершине. Обозначается степень вершины символом $d(x)$. Иногда употребляют термин *локальная степень вершины* или *валентность вершины*.*

На рис. 4.1 изображен граф, степени вершин которого таковы: $d(v_1)=2$; $d(v_2)=3$; $d(v_3)=2$; $d(v_4)=0$; $d(v_5)=1$. Вершина v_4 не инцидентна ни одному ребру и степень её равна 0. Такая вершина называется *изолированной*. Степень вершины v_5 равна 1 (эта вершина инцидентна только одному ребру). Это *висячая* вершина.

Теорема 1. В неориентированном графе без петель сумма степеней всех n вершин графа равна удвоенному числу его рёбер: $\sum_{i=1}^n d(v_i) = 2m$, где $v_i \in V$, $|V|=n$, $|E|=m$.

Теорема 2 (лемма о рукопожатиях). В неориентированном графе без петель количество вершин, имеющих нечетную степень, есть число четное.

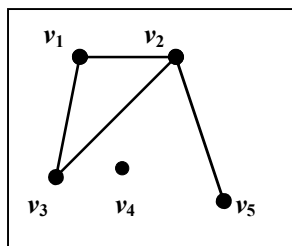


Рис. 4.1

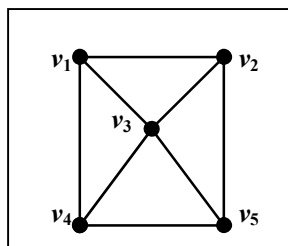


Рис. 4.2

Граф называется *регулярным степени k* (или *однородным*), если степени всех его вершин одинаковы и равны числу k . В таком графе число рёбер $m = nk/2$, что следует из теоремы 2.

Рассмотрим ориентированный граф $G = \langle V, E \rangle$ без петель. Пусть x – какая-либо из его вершин, $x \in V$. Обозначим число дуг, выходящих из вершины через $d^-(x)$. Это число называется *полустепенью исхода*. Число дуг, входящих в вершину, обозначим $d^+(x)$. Это число назовем *полустепенью захода*.

Справедливо следующее равенство: $\sum_{i=1}^n d^-(v_i) + \sum_{i=1}^n d^+(v_i) = 2m$.

Орграф называется *регулярным степени k* , если все его полустепени исхода и захода имеют одно и то же значение: $d^-(x) = d^+(x) = k$. Для такого графа имеет место равенство: $m = nk$.

Задача 4.1. Можно ли построить граф, имеющий одну вершину 4-й степени и три вершины 3-й степени?

Решение. Нельзя. Согласно теореме 2 количество вершин с нечетной степенью должно быть четным. В нашем случае $3+3+3=9$ – число нечетное. А вот если добавить ещё одну вершину 3-й степени, то такой граф будет существовать (рис. 4.2).

5. СПОСОБЫ ЗАДАНИЯ ГРАФОВ

Существует четыре основных базовых метода задания графов. Каждый метод применяется в зависимости от требований конкретной задачи. Однако часто необходимо применять комбинации и модификации этих методов.

1. Матрица инциденций. Данная матрица отражает инцидентность вершин и рёбер графа.

Для неориентированного графа $G = \langle V, E \rangle$, где $|V| = n$, $|E| = m$, это будет матрица H размера $n \times m$, элементы которой определяются так:

$$h_{ij} = \begin{cases} 1, & \text{если ребро } j \text{ инцидентно } i\text{-й вершине;} \\ 0 & \text{– в противном случае.} \end{cases}$$

Для орграфа компоненты матрицы инциденций определяются следующим образом:

$$h_{ij} = \begin{cases} -1, & \text{если дуга выходит из вершины;} \\ 1 & \text{– дуга заходит в вершину;} \\ 0 & \text{– дуга не инцидентна вершине.} \end{cases}$$

Если на вершине v_i существует петля, то в столбце соответствующего ребра (дуги) ставится любое число, отличное от 0, 1 и -1.

Пример. Матрицы инциденций H_1 , H_2 и H_3 для графов, изображенных на рисунках 5.1, 5.2 и 5.3, будут таковы:

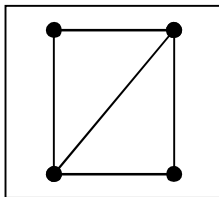


Рис. 5.1

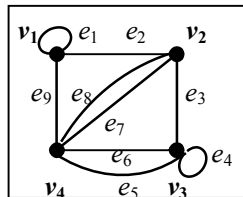


Рис. 5.2

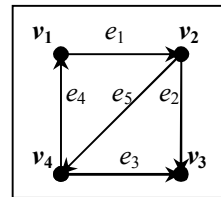


Рис. 5.3

$$\begin{array}{c}
 \begin{array}{c} e_1 \ e_2 \ e_3 \ e_4 \ e_5 \\
 \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \left(\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{array} \right);
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{c} e_1 \ e_2 \ e_3 \ e_4 \ e_5 \ e_6 \ e_7 \ e_8 \ e_9 \\
 \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \left(\begin{array}{ccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right);
 \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \begin{array}{c} e_1 \ e_2 \ e_3 \ e_4 \ e_5 \\
 \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \left(\begin{array}{ccccc} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 \end{array} \right);
 \end{array}
 \end{array}$$

2. Матрица смежностей. Это квадратная матрица M размера $n \times n$, отражающая смежность вершин графа. Коэффициенты этой матрицы определяются следующим образом:

$$m_{ij} = \begin{cases} 1, & \text{если ребро } (v_i, v_j) \in E; \\ 0, & \text{если ребро } (v_i, v_j) \notin E. \end{cases}$$

Если граф без петель, то все диагональные элементы матрицы M равны 0. В случае кратных рёбер между вершинами соответствующий элемент m_{ij} будет равен числу этих рёбер.

Для неориентированного графа матрица смежностей M будет симметричной, то есть $m_{ij} = m_{ji}$ для всех $i, j \leq n$. Для орграфа матрица M , вообще говоря, симметричной не будет.

Пример. Матрицы смежностей M_1 , M_2 и M_3 для графов, изображенных на рисунках 5.1, 5.2 и 5.3, будут следующими:

$$M_1 = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}; \quad M_2 = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 2 & 2 & 0 \end{pmatrix} \end{matrix}; \quad M_3 = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}.$$

3. Списки смежностей. Граф можно представить с помощью списочной структуры, отражающей смежность вершин и состоящей из массива указателей. Для каждой вершины в этом случае нужно составить список вершин, с которыми смежна данная вершина.

Пример. Для графа, изображенного на рис. 5.1, список смежностей таков (рис. 5.4):

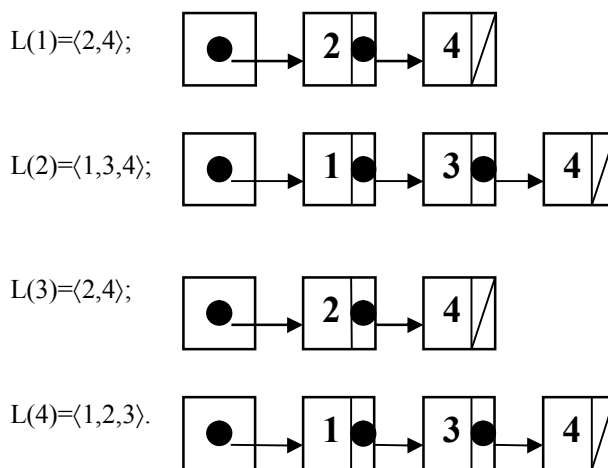


Рис. 5.4

Для орграфа, изображенного на рис. 5.3, список смежностей таков (рис. 5.5):

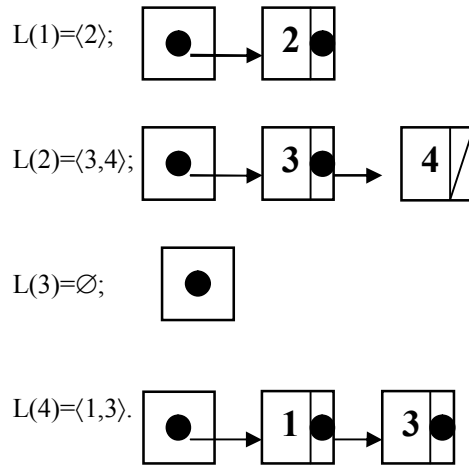


Рис. 5.5

4. Массив (список) рёбер (дуг). Представление графа с помощью массива, отражающего список пар смежных вершин, называется массивом рёбер (для орграфа массив дуг). Каждая строка этого списка соответствует ребру, в ней записаны номера инцидентных ему вершин. Для неориентированного графа порядок вершин произволен, для орграфа первым стоит номер начала дуги, вторым – конца дуги. По этим спискам легко построить матрицу инцидентности графа. Каждая строка списка соответствует столбцу матрицы с тем же номером.

Пример. Для графа (рис.5.1) список рёбер представлен в таблице 5.1, для орграфа (рис.5.3) список дуг в таблице 5.2:

Таблица 5.1

| Ребро | Вершина |
|-------|------------|
| e_1 | v_1, v_2 |
| e_2 | v_2, v_3 |
| e_3 | v_3, v_4 |
| e_4 | v_1, v_4 |
| e_5 | v_2, v_4 |

Таблица 5.2

| Ребро | Вершина |
|-------|------------|
| e_1 | v_1, v_2 |
| e_2 | v_2, v_3 |
| e_3 | v_4, v_3 |
| e_4 | v_4, v_1 |
| e_5 | v_2, v_4 |

Задачи для самостоятельного решения.

1. По заданным матрицам смежности построить неориентированные графы:

$$a) \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}; \quad
 b) \begin{pmatrix} 0 & 2 & 0 & 3 \\ 2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{pmatrix}; \quad
 в) \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 9 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 9 \end{pmatrix}.$$

2. Для заданного орграфа (рис. 5.6) найти матрицы смежности и инцидентности.

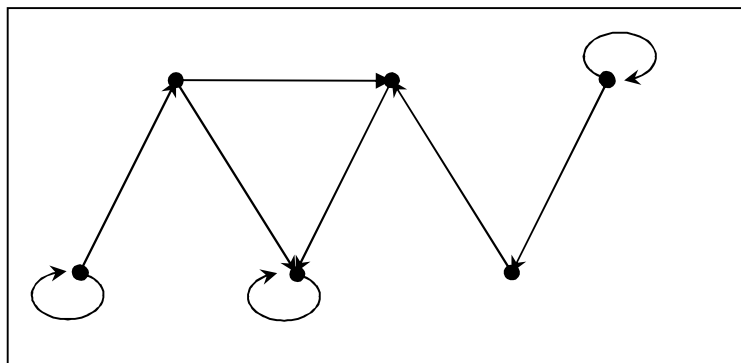


Рис. 5.6

3. По заданным матрицам инцидентности построить неориентированный и ориентированный графы.

$$\text{а) } \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}; \quad \text{б) } \begin{pmatrix} 0 & -1 & -1 & 0 & 9 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{pmatrix}.$$

6. МАРШРУТ, ПУТЬ, ЦЕПЬ, ЦИКЛ

Рассмотрим неориентированный граф $G = \langle V, E \rangle$ без петель и кратных рёбер.

Маршрутом в графе G называется чередующаяся (конечная или бесконечная) последовательность вершин и рёбер $\langle v_1, e_1, v_2, e_2, \dots, v_k \rangle$, в которой любые два соседние элемента инцидентны. Маршрут можно также определить как последовательность смежных вершин $\langle v_1, v_2, \dots, v_k \rangle$, соединённых между собой соответствующими рёбрами e_1, \dots, e_n . Или как последовательность рёбер $\langle e_1, \dots, e_n \rangle$ таких, что любые два соседние имеют общую вершину.

Будем рассматривать лишь конечные последовательности вершин и рёбер. В этом случае вершина v_1 называется начальной, а вершина v_k – конечной. Если начальная и конечная вершины совпадают, то такой путь называют *замкнутым*. В противном случае – *открытым*.

Цепь – это такой маршрут, все рёбра которого разные. *Открытая цепь* – это цепь, у которой начальная и конечная вершина разные. В противном случае цепь называется *замкнутой*.

Путь – это открытая цепь, у которой все вершины разные.

Цикл – это замкнутая цепь, у которой все вершины разные, за исключением первой и последней.

Длина маршрута – это количество рёбер, входящих в него. Очевидно, что количество рёбер всегда будет на 1 меньше количества вершин маршрута.

Если граф не содержит ни одного цикла, его называют *ациклическим*.

Пусть $G = \langle V, E \rangle$ – орграф без петель и кратных дуг. *Ориентированный маршрут* орграфа – это конечная последовательность вершин $\langle v_1, v_2, \dots, v_k \rangle$, где v_1 – начальная, v_k – конечная вершины. Дуги и вершины в маршруте могут появляться несколько раз. Определение замкнутого и открытого маршрута такие же, как для неориентированного графа.

Ориентированный маршрут называют *цепью*, если все его дуги различны. Если в ориентированном маршруте все вершины различны, а цепь открыта, его называют *путём*.

Замкнутая ориентированная цепь называется ориентированным циклом или *контуром*.

Если в орграфе отсутствует контур, то такой граф называют *ациклическим*.

Задача 6.1. Для графа, изображенного на рис. 6.1, указать примеры маршрутов, цепей, путей и циклов. (Для простоты вершины обозначены цифрами).

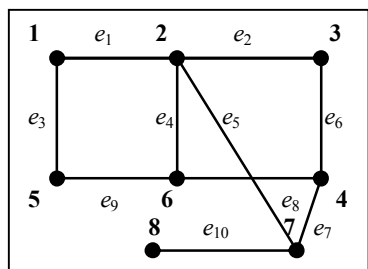


Рис. 6.1

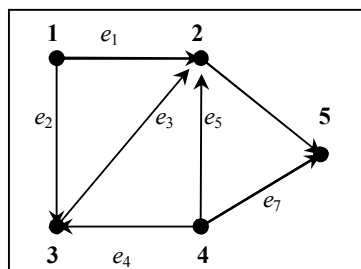


Рис. 6.2

Открытые маршруты:

- 1- e_1 -2- e_2 -3- e_6 -4- e_7 -7- e_{10} -8 (длина маршрута равна 5);
- 1- e_1 -2- e_4 -6- e_8 -4- e_6 -3- e_2 -2- e_5 -7- e_7 -4- e_8 -6- e_9 -5 (длина маршрута равна 9).

Замкнутые маршруты:

- 1- e_1 -2- e_4 -6- e_9 -5- e_3 -1;
- 5- e_3 -1- e_1 -2- e_2 -3- e_6 -4- e_7 -7- e_5 -2- e_4 -6- e_9 -5.

Открытая цепь:

- ✓ 1- e_3 -5- e_9 -6- e_8 -4- e_7 -7- e_5 -2- e_4 -6;
- ✓ 8- e_{10} -7- e_5 -2- e_2 -3- e_6 -4- e_8 -6- e_4 -2- e_1 -1.

Замкнутая цепь:

- ✓ 1- e_3 -5- e_9 -6- e_8 -4- e_7 -7- e_5 -2- e_1 -1;
- ✓ 7- e_5 -2- e_2 -3- e_6 -4- e_7 -7.

Путь:

- ✗ 1- e_1 -2- e_5 -7- e_7 -4- e_8 -6- e_9 -5;
- ✗ 5- e_9 -6- e_8 -4- e_7 -7- e_{10} -8.

Цикл:

- ◆ 1- e_1 -2- e_2 -3- e_6 -4- e_8 -6- e_9 -5- e_3 -1;
- ◆ 2- e_2 -3- e_6 -4- e_7 -7- e_5 -2.

Задача 6.2. Для орграфа, изображенного на рис. 6.2, указать примеры маршрутов, цепей, контуров.

Маршруты:

- 1- e_1 -2- e_6 -5 (длина маршрута равна 2);
- 1- e_2 -3- e_3 -2- e_6 -6 (длина маршрута равна 3).

Все маршруты данного орграфа являются открытыми. Замкнутых маршрутов у него нет.

Цепи. Все рассмотренные маршруты являются цепями, поскольку каждая дуга входит в маршрут только один раз. Кроме того, эти цепи являются также и путями.

Контур. Данный орграф не имеет контуров. Он является ациклическим.

Задачи для самостоятельного решения.

1. По заданным матрицам смежности построить графы и привести примеры маршрутов, цепей и т.д. Определить их длины.

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}; \quad M_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

7. СВЯЗНОСТЬ ГРАФОВ

Рассмотрим неориентированный граф $G = \langle V, E \rangle$. Две вершины v_i и v_j называют *связными* в графе G , если в нём существует путь из v_i к v_j . Каждая вершина графа считается связной по отношению к себе самой.

Граф называют *связным*, если в нём существует путь для каждой пары его вершин. На рис. 7.1 изображен связный граф.

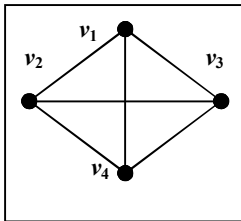


Рис. 7.1

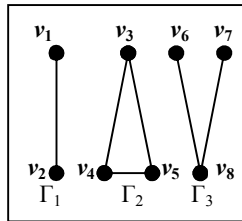


Рис. 7.2

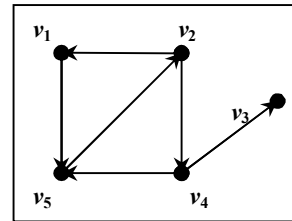


Рис. 7.3

Рассмотрим граф, изображенный на рис. 7.2. Это пример несвязного графа. Множество вершин такого графа можно разбить на подмножества так, чтобы графы, порождённые этими подмножествами, были бы связными. При этом ни одна вершина одного подмножества не связана ни с одной вершиной любого другого подмножества вершин. Каждая такая часть несвязного графа называется компонентой связности данного графа. Так Γ_1 , Γ_2 и Γ_3 — это и есть компоненты связности данного графа, которые порождены подмножествами вершин графа $V_1 = \{v_1, v_2\}$, $V_2 = \{v_3, v_4, v_5\}$, $V_3 = \{v_6, v_7, v_8\}$.

Если граф связный, то он имеет единственную компоненту связности, которой является он сам.

Рассмотрим орграф $G = \langle V, E \rangle$. Отношение связности вершин в орграфе не является симметричным, в отличие от неориентированного графа, где две вершины либо связаны, либо нет. Поэтому определение связности в орграфе несколько иное.

Пусть v_i и v_j — вершины орграфа. Тогда:

1. две вершины v_i и v_j орграфа G *сильно связны*, если существует ориентированная цепь из v_i в v_j и из v_j в v_i ;
2. две вершины v_i и v_j орграфа G *односторонне связны*, если существует путь или из v_i в v_j , или наоборот;
3. две вершины v_i и v_j орграфа G *слабо связны*, если они будут связными в графе, полученном из данного заменой всех дуг на дуги противоположной ориентации.

Если все вершины в орграфе *сильно (односторонне, слабо)* связны, то такой граф называется *сильно (односторонне, слабо)* связным. Сильная связность графа влечёт его одностороннюю связность, которая, в свою очередь, влечёт слабую связность. Обратное утверждение неверно.

На рис. 7.3 изображен орграф. Вершины v_1 и v_4 – сильно связны, так как существует путь из v_1 в v_4 и обратно: $v_1 - v_5 - v_2 - v_4$ и $v_4 - v_5 - v_2 - v_1$. Вершины v_1 и v_3 – односторонне связаны, так как существует путь только из v_1 в v_3 : $v_1 - v_5 - v_2 - v_4 - v_3$. Слабо связных вершин на данном орграфе нет. Этот граф является односторонне связным.

8. ПОДГРАФ. ЧАСТИ ГРАФА

Рассмотрим графы $G = \langle V, E \rangle$ и $G' = \langle V', E' \rangle$. Пусть $V' \subset V$ и $E' \subset E$, а ребро $(v_i', v_j') \in E'$ тогда и только тогда, когда оба его конца включены в множество V' ($v_i' \in V'$ и $v_j' \in V'$). В этом случае граф G' называют *собственным подграфом* графа G .

Из этого определения следует, что для образования подграфа множество его вершин V' можно выбрать произвольно, в то время как множество рёбер однозначно определяется выбором его вершин.

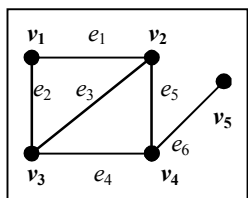


Рис. 8.1

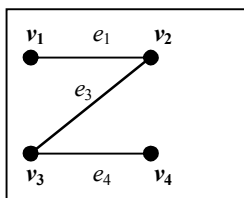


Рис. 8.2

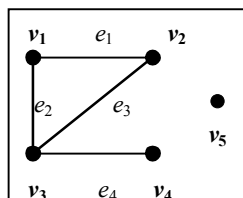


Рис. 8.3

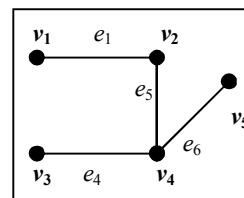


Рис. 8.4

На рис. 8.1 изображен граф G , имеющий 5 вершин и 6 рёбер. На рис. 8.2 изображён его подграф G' , у которого 4 вершины (которые являются подмножеством вершин графа G) и 3 ребра (которые также есть подмножество рёбер G). На рис. 8.3 показан еще один подграф графа G с изолированной вершиной.

Остовным графом графа G (или просто *остовом*) называется такой связный граф G' , все вершины которого являются вершинами G ($V'=V$) и который не содержит циклов. На рис. 8.4 показан остовный граф графа G , изображённого на рис. 8.1.

Звездой вершины v_i графа G называется часть графа, состоящая из данной вершины и всех инцидентных ей рёбер со смежными вершинами. На рис. 8.5 показана звезда вершины v_2 .

Дополнением графа G называют граф $\bar{G} = \langle V, E' \rangle$, если он имеет то же множество вершин, а две его вершины смежны тогда и только тогда, когда они не смежны в G . То есть ребро $(v_i, v_j) \in E'$ тогда, когда $(v_i, v_j) \notin E$. На рис. 8.6 показано дополнение графа G .

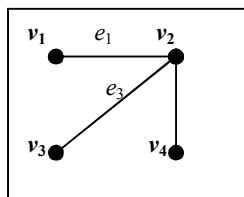


Рис. 8.5

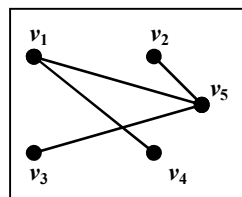


Рис. 8.6

Задачи для самостоятельного решения.

1. Найти собственные подграфы графа G (рис. 8.5), его дополнение и остовы.

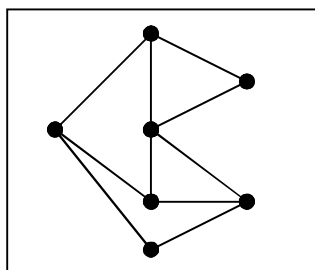


Рис. 8.5

9. ДЕРЕВО И ЛЕС

Графы, которые встречаются во многих прикладных задачах, являются связными. Среди этих графов существуют такие, которые имеют наиболее простую структуру. Этот класс графов называют деревьями.

Неориентированное дерево – это связный неориентированный граф, который не имеет циклов (ациклический). Простейшим примером дерева могут служить *цепь* и *звезда* (рис. 9.1 и рис. 9.2). На рис. 9.3 изображен типичный пример дерева.

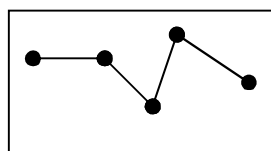


Рис. 9.1

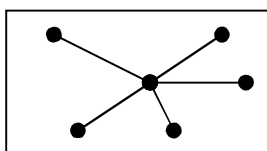


Рис. 9.2

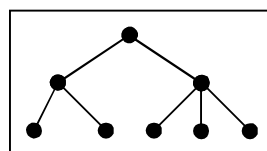


Рис. 9.3

Можно дать другое определение остова графа, основанное на понятии дерева. *Остов* графа G – это его дерево, которое содержит все вершины данного графа.

Графы типа дерево обладают некоторыми специфическими свойствами:

1. В любом дереве существуют, по крайней мере, две концевые вершины – висящие степени 1.
2. Между двумя любыми вершинами дерева существует ровно один путь.
3. Дерево с n вершинами содержит ровно $n-1$ ребро.

Лесом называется несвязный граф, не имеющий циклов. Очевидно, что компонентами связности леса будут деревья (рис. 9.4).

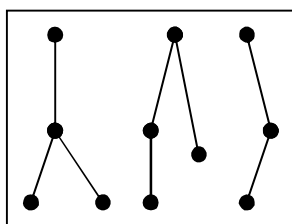


Рис.9.4

Для неориентированных деревьев имеет место следующая теорема, которая носит имя английского математика А. Кэли.

Теорема Кэли. На n вершинах можно построить n^{n-2} различных деревьев.

Эта теорема применяется при постановке и решении класса задач о минимальном соединении. Пусть имеем следующую задачу: как соединить n городов дорогами так, чтобы не строить лишних, и сколько существует способов построения такой сети дорог?

Поставим в соответствие множеству V вершин графа множество городов. Положим, что известна стоимость сооружения дороги между каждыми двумя пунктами v_i и v_j , которую обозначим $\mu(v_i, v_j)$. Каждому дереву на n вершинах ставим в соответствие сумму $\sum_{(v_i, v_j) \in E} \mu(v_i, v_j)$, равную стоимости строительства соответствующей сети. Рассмотрев, согласно теореме Кэли, не более n^{n-2}

деревьев и соответствующих им сумм, выберем из них минимальную, тем самым получим решение задачи.

Ориентированное дерево (ордереве или корневое дерево) – это граф, обладающий следующими свойствами:

1. Существует единственная вершина, называемая *корнем*, в которую не входит ни одна дуга.
2. В каждую вершину (кроме корня) заходит только одна дуга.
3. Каждая вершина достижима из корня, то есть существует ориентированный путь из корня в каждую вершину орграфа.

На рис. 9.5 показаны различные примеры ориентированных деревьев.

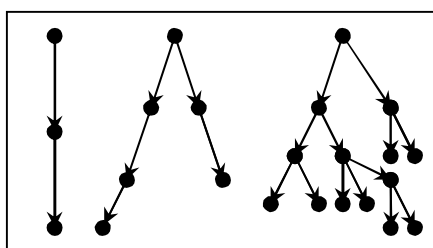


Рис. 9.5

Конечная вершина ордерова называется *листом*. Путь из корня к листу называют *ветвью*. Длина наибольшей ветви ордерова называют его *высотой*. *Уровень точки* (вершины) ордерова – это длина от корня до данной точки. Уровень самого корня равен 0. Точки одного уровня образуют *ярус* дерева.

Ориентированные деревья являются абстракцией иерархических отношений. В программировании интенсивно используются ордеревья для построения алгоритмов, а структура вложенных каталогов и файлов в современных операционных системах находит отображение, например, в термине *корневой каталог дисков*.

Задача 9.1. Построить деревья графа с n вершинами (если $n = 2,3$).

Решение. Согласно теореме Кэли количество деревьев: для $n = 2$ равно $2^{2-2} = 2^0 = 1$ (рис. 9.6); для $n = 3$ равно $3^{3-2} = 3$ (рис. 9.7).

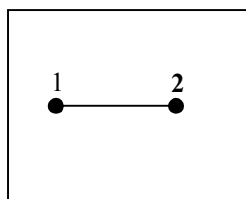


Рис. 9.6

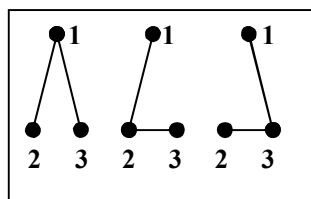


Рис. 9.7

Задачи для самостоятельного решения.

1. Построить все деревья на четырёх вершинах.

10. ИЗОМОРФИЗМ ГРАФОВ

Один и тот же граф можно изобразить различными схемами, поскольку имеется большая свобода при размещении вершин и в выборе формы соединяющих их рёбер (дуг). Рассмотрим граф G_1 (рис. 10.1) и граф G_2 (рис. 10.2). Что можно сказать об этих изображениях? Визуально они различны, хотя и представляют собой изображение одного и того же графа.

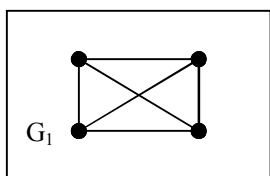


Рис.10.1

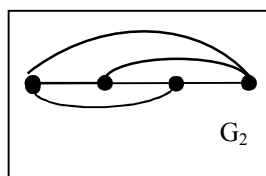


Рис.10.2

Как же на практике можно определить, какие графы различаются, а какие нет? Для этого и вводится понятие *изоморфизма* графов. В переводе с греческого «изоморфизм» означает «одинаковой формы».

Графы $G=\langle V, E \rangle$ и $G'=\langle V', E' \rangle$ называются *изоморфными*, если существует такое взаимно однозначное соответствие (биекция) между множествами их вершин V и V' , что вершины в одном из графов соединены рёбрами в том и только в том случае, когда соответствующие им вершины соединены рёбрами в другом графе. Иными словами, множества рёбер E и E' сохраняют отношение инцидентности: если $v \in V$, $v' \in V'$ и $e \in E$, $e' \in E'$, то вершина v' инцидентна ребру E' тогда и только тогда, когда вершина v инцидентна ребру E .

Графы, имеющие различное число вершин или рёбер, не могут быть изоморфными. Числовые характеристики графов – количество вершин n и количество рёбер t для изоморфных графов есть инварианты.

В случае *изоморфизма ориентированных* графов ко всему рассмотренному выше следует добавить ещё одно требование: направления соответствующих дуг должны совпадать.

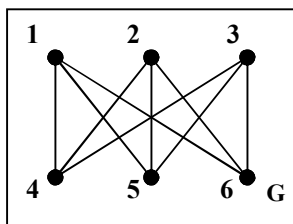


Рис. 10.3

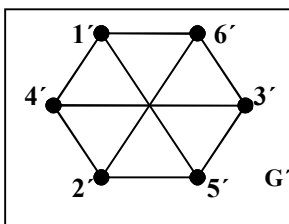


Рис. 10.4

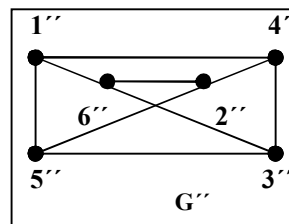


Рис. 10.5

На рисунках 10.3, 10.4 и 10.5 показаны диаграммы трёх изоморфных графов G , G' и G'' с инвариантами $n=6$, $m=9$. В таблице 10.1 представлены списки смежности вершин для этих графов.

Таблица 10.1

| ГРАФ G | | ГРАФ G' | | ГРАФ G'' | |
|----------|--------------------|-----------|--------------------|------------|--------------------|
| вершина | смежна вершинам | вершина | смежна вершинам | вершина | смежна вершинам |
| 1 | 4, 5, 6 | 1' | 4', 5', 6' | 1'' | 4'', 5'', 6'' |
| 2 | 4, 5, 6 | 2' | 4', 5', 6' | 2'' | 4'', 5'', 6'' |
| 3 | 4, 5, 6 | 3' | 4', 5', 6' | 3'' | 4'', 5'', 6'' |
| 4 | 1, 2, 3 | 4' | 1', 2', 3' | 4'' | 1'', 2'', 3'' |
| 5 | 1, 2, 3 | 5' | 1', 2', 3' | 5'' | 1'', 2'', 3'' |
| 6 | 1, 2, 3 | 6' | 1', 2', 3' | 6'' | 1'', 2'', 3'' |

Граф G^* , схема которого изображена на рис. 10.6, не будет изоморфен графам G и другим. Хотя он и имеет те же характеристики ($n=6$, $m=9$), но отсутствует биекция между множествами V и V^* . Так вершина 2^* графа G^* смежна вершинам 3^* , 4^* и 6^* , в то время как в графе G вершина 2 смежна вершинам 4, 5 и 6.

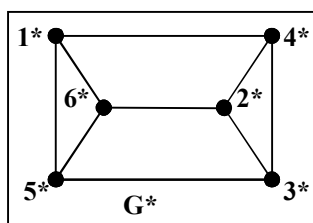


Рис. 10.6

Таким образом, изоморфные графы несут одну и ту же информацию, поэтому они могут рассматриваться как один граф. Графы различаются с точностью до изоморфизма.

Задачи для самостоятельного решения.

1. Являются ли изоморфными графы, представленные на рис. 10.1 и рис. 10.7?

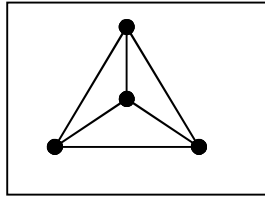


Рис. 10.7

11. ПЛАНАРНОСТЬ ГРАФОВ

Рассмотрим неориентированный граф $G = \langle V, E \rangle$. Граф называется *плоским*, если все его вершины есть точки плоскости, а рёбра – непрерывные плоские линии без самопересечений такие, что никакие два ребра не имеют общих точек, кроме инцидентной им обоим вершины. Примеры плоских графов представлены на рис. 11.1, неплоских – на рис. 11.2.

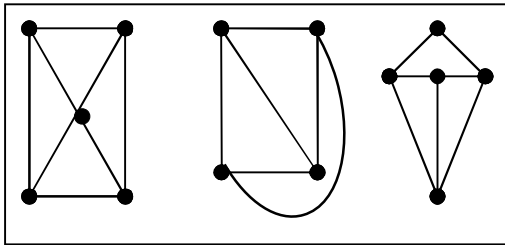


Рис. 11.1

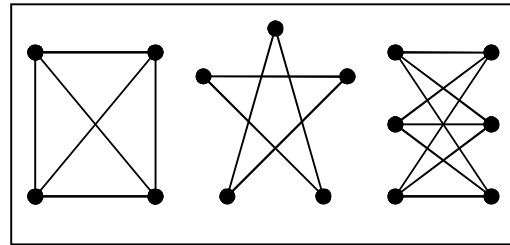


Рис. 11.2

Таким образом, у плоского графа никакие два ребра не пересекаются между собой нигде, кроме вершины.

Произвольный граф, который изоморфен плоскому, называется *планарным графом*.

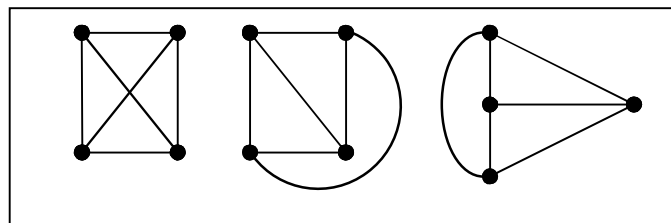


Рис.11.3

На рисунке 11.3 граф G не является плоским. Однако он изоморфный графам G' и G'' , которые являются плоскими. Следовательно, граф G – планарный.

Процесс изоморфного преобразования графа, в результате которого он становится плоским, называется *укладкой* графа. Задачи укладки имеют широкое применение на практике, например, при разработке информационных, транспортных и энергетических сетей, при создании микросхем, плат и чипов.

Одной из первых задач по укладке графов была старинная шуточная «задача о трёх колодцах», которая формулируется следующим образом: «Три поссорившихся друг с другом соседа, живущих каждый в своём доме, имеют три общих колодца. Можно ли провести непересекающиеся дорожки от каждого дома к каждому колодцу?»

Задача имеет отрицательное решение. Доказательство принадлежит французскому математику К. Жордану, который доказал, на первый взгляд, очевидную теорему, которая названа его именем.

Теорема Жордана. Если L – непрерывная плоская кривая без самопересечений на плоскости, то она делит плоскость на внешнюю и внутреннюю области так, что любая непрерывная линия, соединяющая две точки x и y из внутренней и внешней области, пересекает L (рис. 11.4).

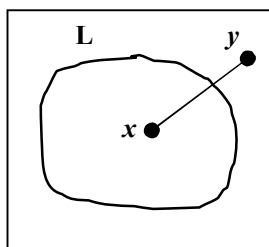


Рис. 11.4

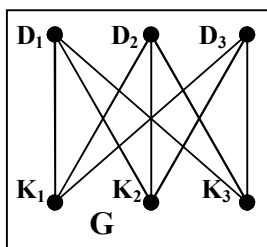


Рис. 11.5

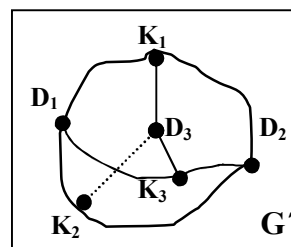


Рис. 11.6

Для задачи «о трёх колодцах» рассмотрим граф G (рис.11.5), соответствующий взаимному расположению домов и колодцев. Здесь вершины, соответствующие домам, обозначены D_i , колодцам – K_j . Построим граф G' , изоморфный данному (рис. 11.6). Из теоремы Жордана следует, что соединение вершин D_3 и K_2 без пересечения с замкнутой кривой, на которой располагаются вершины K_1, D_2, K_3 и D_1 , невозможно.

Если сделать попытку перенумеровать вершины графа, чтобы получить какое-либо другое решение, то, в силу сохранения инцидентности этих вершин рёбрам, получим графы, изоморфные первоначальному. Таким образом и в общем случае имеем отрицательное решение. Граф, соответствующий задаче «о трёх колодцах» (рис. 11.5), является непланарным. В силу своей важности, он имеет специальное обозначение A_{33} в теории графов.

12. ДВУДОЛЬНЫЕ ГРАФЫ

Граф $G=\langle V, E \rangle$ называется *двудольным* (или *биграфом*), если множество его вершин V можно разбить на два непересекающихся множества V_1 и V_2 ($V_1 \cap V_2 = \emptyset$) так, что каждое ребро из множества E будет инцидентно одной вершине из V_1 и другой вершине из множества V_2 (рис. 12.1). Множества V_1 и V_2 называют *долями двудольного графа* G . Если двудольный граф содержит все рёбра, соединяющие вершины из множеств V_1 и V_2 , то его называют *полным двудольным графом*. Иногда для двудольного графа используют специальное обозначение $G=\langle V_1, V_2, E \rangle$.

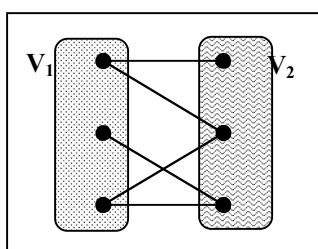


Рис. 12.1

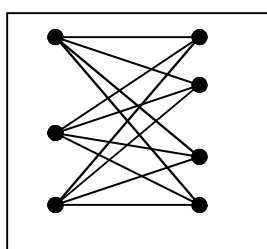


Рис. 12.2

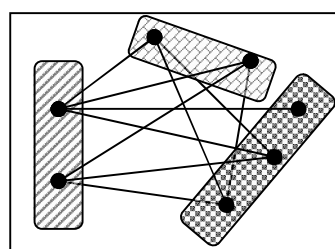


Рис. 12.3

Для полного двудольного графа существует специальное обозначение. Если $|V_1|=m$, $|V_2|=n$, то такой граф обозначается символом $K_{m,n}$. На рис. 12.2 изображена диаграмма полного двудольного графа $K_{3,4}$.

Признак двудольности графа. Граф является двудольным в том и только в том случае, когда он не содержит циклов нечётной длины.

Кроме рассмотренных двудольных существуют также n -дольные графы, которые определяются подобным образом. Однако нет простых способов распознавания n -дольных графов при $n > 2$. Диаграмма трёхдольного графа показана на рис. 12.3.

Двудольные графы применяются во многих задачах дискретной математики. Например, с их помощью изображались соответствия на различных множествах (см. часть I настоящего пособия).

13. ПАРОСОЧЕТАНИЯ В ГРАФЕ

Рассмотрим так называемую *теорему о свадьбах*, доказанную Ф. Холлом в 1935 г., которая отвечает на следующий вопрос, известный под названием *задачи о свадьбах*: «Рассмотрим некоторое конечное множество юношей, каждый из которых знаком с несколькими девушками. Спрашивается, при каких условиях можно женить юношей так, чтобы каждый из них женился на знакомой ему девушке? (Будем считать, что полигамия не разрешена)».

Например, пусть имеется четверо юношей $\{b_1, b_2, b_3, b_4\}$ и пять девушек $\{g_1, g_2, g_3, g_4, g_5\}$, а отношения знакомства между ними показаны в таблице 13.1.

Таблица 13.1

| юноша | девушки, с которыми знаком юноша |
|-------|----------------------------------|
| b_1 | g_1, g_4, g_5 |
| b_2 | g_1 |
| b_3 | g_2, g_3, g_4 |
| b_4 | g_2, g_4 |

Эту задачу можно представить графически, взяв двудольный граф G с множеством вершин, разделенных на два непересекающихся подмножества V_1 и V_2 , представляющих юношей и девушек, соответственно, соединив ребром каждого юношу со знакомой ему девушкой (рис. 13.1).

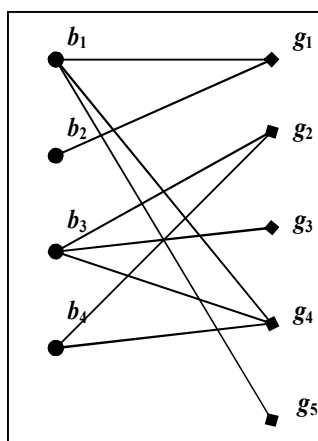


Рис. 13.1

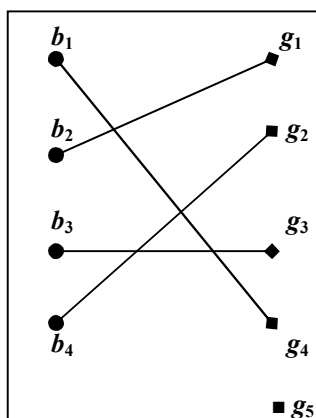


Рис.13.2

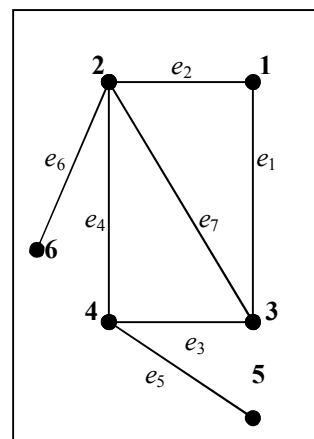


Рис. 13.3

Тогда возможно следующее решение: b_1 женится на g_4 , b_2 – на g_1 , b_3 – на g_3 , а b_4 – на g_2 (рис. 13.2).

Рассмотренная задача приводит к понятию *паросочетаний* графа.

Пусть задан некоторый (не обязательно двудольный) граф $G = \langle V, E \rangle$. Два ребра графа называются *независимыми*, если они не имеют общей вершины (то

есть не являются смежными). Рёбра e_1, e_2, \dots, e_k называются независимыми, если каждая пара из них не имеет общей вершины. *Паросочетанием* в графе G называется множество M независимых (несмежных) рёбер. *Размером* паросочетания называется число его рёбер. Очевидно, что размер паросочетания не что иное, как мощность множества M .

Максимальное паросочетание – то, которое содержит максимальное число рёбер. Очевидно, что его размер нельзя увеличить присоединением какого-либо ребра графа. Наибольшее паросочетание всегда является максимальным. Число рёбер в максимальном паросочетании называют *числом паросочетаний* графа и обозначают символом $\alpha(G)$.

Рассмотрим паросочетания графа, изображенного на рис. 14.3: $M_1=\{e_2, e_3\}$, $M_2=\{e_2, e_5\}$, $M_3=\{e_5, e_7\}$, $M_4=\{e_1, e_4\}$, $M_5=\{e_1, e_5, e_6\}$. Максимальным будет паросочетание M_5 , содержащее три ребра. Поэтому число паросочетаний графа G (рис. 13.2) будет равно $\alpha(G)=|M_5|=3$.

Вершина графа, инцидентная некоторому ребру из паросочетания M , называется *насыщенной*. В противном случае такая вершина называется *ненасыщенной*. Паросочетание, в котором все вершины графа насыщены, называют *совершенным*. Совершенное паросочетание всегда является наибольшим. Обратное утверждение, вообще говоря, неверно.

У графа G (рис. 13.3) насыщенными будут вершины 1, 2, 3 и 4 в паросочетании M_1 , в M_2 – вершины 1, 2, 4, 5 и т.д. Совершенным же будет паросочетание M_5 , которое насыщает все вершины графа.

Совершенным паросочетанием из V_1 в V_2 в *двудольном графе* называется взаимно однозначное соответствие между вершинами из V_1 и подмножеством вершин из V_2 , обладающее тем свойством, что соответствующие вершины соединены ребром. В этом случае каждая вершина из V_1 является насыщенной. На рис. 13.2 показана схема такого двудольного графа. Этот же граф иллюстрирует решение задачи о свадьбах.

Тогда *задачу о свадьбах* можно выразить в терминах теории графов следующим образом: если G — двудольный граф, то при каких условиях в G существует совершенное паросочетание из V_1 в V_2 ?

Можно сформулировать следующее очевидное утверждение: необходимое условие для существования решения задачи о свадьбах заключается в том, что любые k юношей из данного множества должны быть знакомы (в

совокупности), по меньшей мере, с k девушками (для всех целых k , удовлетворяющих неравенствам $1 \leq k \leq m$, где через m обозначено общее число юношей). Необходимость этого условия сразу вытекает из того, что если оно не верно для какого-нибудь множества юношей, то мы не сможем женить требуемым способом даже этих юношей, не говоря уже об остальных. Это очевидное необходимое условие является в то же время и достаточным. В этом и состоит теорема Холла о свадьбах.

Одним из многочисленных приложений теоремы Холла являются так называемые латинские квадраты.

Рассмотрим прямоугольную матрицу, обладающую тем свойством, что все её элементы в каждой строке и в каждом столбце различны.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 5 & 2 & 1 & 4 \end{pmatrix}$$

Такая матрица носит название *латинского прямоугольника*. Возникает вопрос: а можно ли к данному прямоугольнику присоединить некоторое число строк, чтобы получилась квадратная матрица (латинский квадрат)? Оказывается, это всегда возможно.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 5 & 2 & 1 & 4 \\ 4 & 3 & 5 & 2 & 1 \\ 5 & 1 & 4 & 3 & 2 \end{pmatrix}$$

Латинские квадраты долгое время были известны лишь математикам и любителям головоломок и, в основном, благодаря одной знаменитой задаче Л. Эйлера. В 1782 г. он предложил следующую проблему: среди 36 офицеров находится по шесть офицеров шести различных званий из шести полков. Можно ли построить этих офицеров в каре так, чтобы в каждой колонне и каждой шеренге встречались офицеры всех званий и всех полков?

Лишь в 1901 г. удалось доказать, что это невозможно. Однако связанные с задачей Эйлера латинские квадраты не потеряли интереса, так как вскоре обнаружилось, что они имеют многообразные практические применения. А в конце 60-х годов двадцатого столетия они были применены в теории кодирования. Получающиеся на их основе коды допускают простые алгоритмы декодирования.

14. БИНАРНЫЕ ОТНОШЕНИЯ И ГРАФЫ

Пусть на некотором непустом множестве A мощности n задано отношение $R \subset A \times A = \{(a, b) \mid a \in A, b \in A, aRb\}$. Отношение R – это некоторое подмножество декартова произведения $A \times A$, то есть множество упорядоченных пар элементов множества A .

Отождествим множество A с множеством V вершин некоторого графа G . Тогда множество упорядоченных пар (a, b) можем рассматривать как множество E рёбер этого же графа. Таким образом всякое отношение на множестве A (оно же V) есть некоторый граф $G = \langle V, E \rangle$.

Обратное утверждение, как правило, неверно, поскольку не всякий граф можно трактовать как бинарное отношение. Например, если граф имеет кратные рёбра, то эту неоднозначность нельзя истолковать как отношение. В остальном же связь между графами и бинарными отношениями является достаточно тесной. Фактически это один и тот же класс объектов, которые описаны по-разному. Графы дают возможность наглядно изображать модели задач в виде диаграмм, что достаточно важно при проектировании и программировании.

Пусть имеем граф $G = \langle V, E \rangle$, порождённый отношением R на множестве V . Если R является *рефлексивным* (то есть aRa для всех $a \in V$), то множество E будет включать петли для каждой вершины. Если R *симметрично* (то есть из aRb следует bRa для всех $a \in R, b \in R$), то каждому ребру (a, b) будет соответствовать ребро противоположной направленности (b, a) , что может быть эквивалентно неориентированному графу. Если R *транзитивно* (то есть из aRb и bRc следует, что aRc), то множество E наряду с дугами (a, b) и (b, c) будет содержать и их замыкание – дугу (a, c) . Иными словами, если есть путь, то есть и дуга.

Если R обладает свойствами рефлексивности, симметричности и транзитивности, то это есть отношение *эквивалентности*. Такое отношение разбивает множество на классы эквивалентности. Элементы из одного класса находятся в этом отношении, элементы же из разных классов – нет.

Граф, соответствующий отношению эквивалентности – это неориентированный граф, состоящий из нескольких подграфов, являющихся компонентами его связности. Каждый такой подграф есть полный граф с

петлями и соответствует определённому классу эквивалентности отношения R (рис. 14.1).

Рассмотрим отношение *порядка*, которое характеризуется отсутствием симметричности и наличием транзитивности. Здесь различают отношения *строгого* и *нестроого* порядков.

Строгий порядок – антирефлексивность, асимметричность и транзитивность. Соответствующий этому отношению оргграф будет без петель.

Нестрогий порядок – рефлексивность, антисимметричность и транзитивность. Соответствующий оргграф будет иметь петли.

Кроме того, множество, на котором определено отношение порядка, может быть упорядочено *линейно (полностью)* или *частично* этим отношением.

Линейная (полная) упорядоченность, когда выполняется отношение либо aRb , либо bRa . Соответствующий оргграф таков:

- Для строгого порядка – любые две вершины графа соединены дугой, и для каждой пары дуг, инцидентных одной и той же вершине, определена замыкающая их транзитивная дуга. Петель нет (рис. 14.3).
- Для нестроого порядка – оргграф отличается от предыдущего наличием петель (рис. 14.4).

Частичная упорядоченность – когда отношение aRb определено не для всех пар (a, b) , а лишь для некоторых. Соответствующий оргграф таков:

- Для строгого порядка – не всякие две его вершины соединены дугой, но если существует цепь между тремя какими-либо вершинами, то обязательно есть и замыкающая её транзитивная дуга. Петель нет (рис. 14.5).
- Для нестроого порядка – оргграф отличается от предыдущего наличием петель (рис. 14.6).

Задача 14.1. На множестве $A = \{9, 13, 18, 21, 30\}$ определено отношение $R = \{(a, b) \mid a \in A, b \in A, a-b \text{ делится нацело на } 4\}$.

Решение. Построим матрицу C_R данного отношения:

$$C_R = \begin{matrix} & \begin{matrix} 9 & 13 & 18 & 21 & 30 \end{matrix} \\ \begin{matrix} 9 \\ 13 \\ 18 \\ 21 \\ 30 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

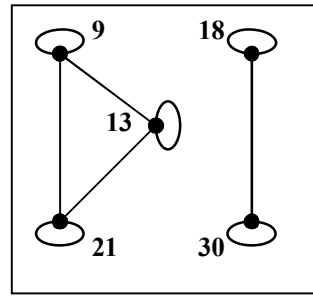


Рис. 14.1

Отношение является рефлексивным, симметричным и транзитивным. Следовательно – это *отношение эквивалентности*. Классы эквивалентности: $K(9)=K(13)=K(21)= \{9, 13, 21\}$; $K(18)=K(30)=\{18, 30\}$. Схема соответствующего графа показана на рис. 14.1.

Задача 14.2. На множестве отрезков задано отношение R : отрезок x короче отрезка y .

Решение. Рассмотрим множество $A=\{a, b, c, d, e\}$, состоящее из пяти различных отрезков (рис. 14.2), и построим матрицу C_R данного отношения.

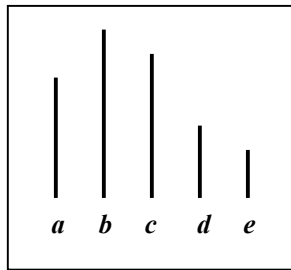


Рис. 14.2

$$C_R = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

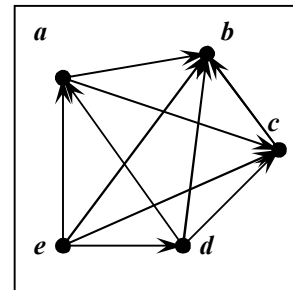


Рис. 14.3

Данное отношение антирефлексивно, асимметрично и транзитивно. Это отношение строгого порядка, которое упорядочивает множество A линейно: если отрезок x меньше отрезка y (xRy), то отрезок y уже не может быть меньше отрезка x (yRx не выполняется). Данное отношение является отношением строгого порядка, которое линейно упорядочивает множество A . Граф этого отношения показан на рис. 14.3.

Задача 14.3. На множестве углов $A = \{30^\circ, 45^\circ, 75^\circ, 90^\circ, 135^\circ\}$ задано отношение $R = \{(\alpha, \beta) \mid \alpha \in A, \beta \in A, \angle \alpha \text{ не больше } \angle \beta\}$. Построить граф данного отношения.

Решение. Построим матрицу данного отношения. Свойства отношения: рефлексивность, антисимметричность и транзитивность. Это отношение

нестроого порядка. Множество A является линейно упорядоченным данным отношением (выполняется одно из отношений: либо $\alpha R \beta$, либо $\beta R \alpha$).

Данное отношение – отношение *нестроого* порядка, которое *линейно упорядочивает* множество A . Граф отношения представлен на рис. 14.4.

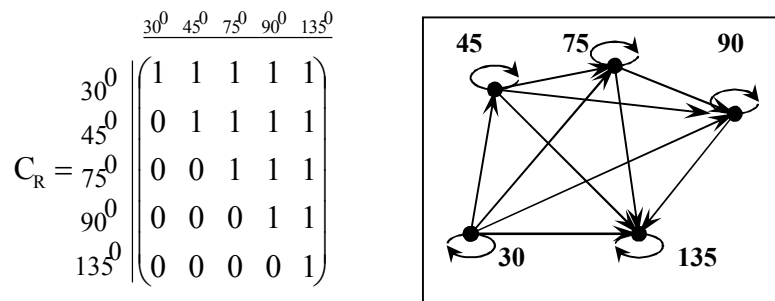


Рис. 14.4

Задача 14.4. Рассмотрим отношение «быть подчинённым» на примере некоторой небольшой организации. Пусть в первом отделе работает три сотрудника (a, b, c), которые подчинены начальнику этого отдела (m). Во втором отделе – два сотрудника (d, e), которые подчинены своему начальнику отдела (n). Все работающие находятся в прямом подчинении управляющего (u). Таким образом, получаем следующую задачу: на множестве $A = \{a, b, c, d, e, m, n, u\}$ рассмотреть отношение $R = \{(x, y) \mid x \in A, y \in A, x \text{ подчинён } y\}$.

Решение. Построим матрицу C_R . Данное отношение R является отношением строого порядка (оно антирефлексивно, асимметрично и транзитивно) и упорядочивает множество A частично, поскольку для пар элементов (a, b) , (a, c) , (b, c) , (a, d) , (m, n) и т. д. порядок «подчинённости» не определён. Поэтому между соответствующими вершинами нет дуг. Данное отношение – отношение *строого* порядка, которое *частично упорядочивает* множество A . Его граф показан на (рис. 14.5).

$$C_R = \begin{matrix} & \begin{matrix} a & b & c & d & e & m & n & u \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ m \\ n \\ u \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

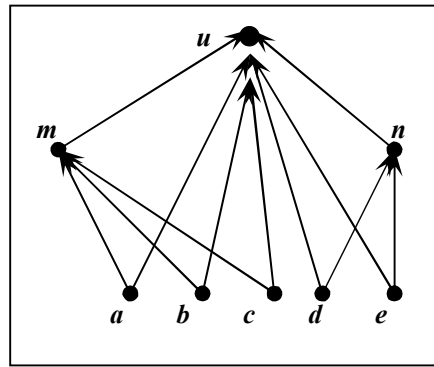


Рис. 14.5

Задача 14.5. На множестве $A = \{2, 3, 6, 7, 11, 14\}$ задано отношение $R = \{(m,n) \mid m \in A, n \in A, m \text{ делитель } n\}$. Построить граф этого отношения.

Решение. Перечислим упорядоченные пары отношения: $R = \{(2,6), (2,14), (3,6), (7,14), (2,2), (3,3), (6,6), (7,7), (11,11), (14,14)\}$. Свойства отношения: рефлексивность ((2,2), (3,3) и т. д.), антисимметричность (есть (2,6), но нет (6,2) и т. д.), транзитивность (из (2,6) и (6,6) следует (2,6); из (2,14) и (14,14) следует (2,14) и т. д.). Это отношение нестрогого порядка, которое упорядочивает множество A частично (выполняется $2R6$ и не выполняется $6R2$; выполняется $2R14$ и не выполняется $14R2$; не выполняется ни (2,3), ни (3,2); не выполняется ни (2,7), ни (7,2) и т. д.).

Данное отношение – отношение *нестрогого порядка*, которое *частично упорядочивает* множество A . Схема графа представлена на рис. 14.6.

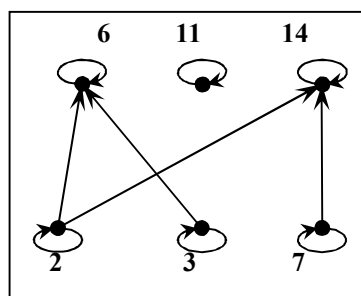


Рис. 14.6

Задачи для самостоятельного решения.

1. Определить, какими свойствами обладают данные бинарные отношения? Изобразить их графически.

$$\begin{array}{cccc}
 a) \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}; &
 b) \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}; &
 c) \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}; &
 d) \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.
 \end{array}$$

2. Является ли $R = \{(a,b) \mid a \in A, b \in A, a + b - \text{четное}\}$ отношением эквивалентности на множестве $A = \{1, 2, 3, 5, 8\}$. Изобразить R графически.
3. На множестве треугольников задано отношение: $\triangle ABC$ подобен $\triangle A_1B_1C_1$. Показать, что это отношение эквивалентности и изобразить его графически.
4. На множестве городов Украины рассмотреть отношение: город a расположен восточнее города b . Определить свойства этого отношения и изобразить графически.
5. Рассмотреть отношение «быть подмножеством ($A \subseteq B$) в булеане некоторого множества» на примере множества $S = \{1, 2, 3\}$. Определить свойства этого отношения и изобразить его графически.

15. ЭЙЛЕРОВЫ ГРАФЫ

Вернёмся к задаче о кенигсбергских мостах (см. разд. 1). Задачу о нахождении маршрута через мосты Эйлер свёл к нахождению цикла на графе и показал, что такового не существует. А существуют ли вообще такие графы, которые имеют цикл, проходящий через каждое ребро графа ровно один раз? Оказывается, существуют. И такие графы были названы эйлеровыми.

Рассмотрим неориентированный граф $G = \langle V, E \rangle$. *Эйлеров путь* на графе – это путь, проходящий через каждое ребро в точности один раз. Если начальная и конечная точки такого пути совпадают, то такой цикл называется *эйлеровым циклом*. Граф, содержащий эйлеров цикл, называется *эйлеровым графом*.

Очевидно, что не всякий граф содержит такой цикл. Каковы же критерии существования подобных путей и графов?

Теорема. В конечном неориентированном графе $G = \langle V, E \rangle$ эйлеров путь существует тогда и только тогда, когда:

- 1) граф связан;
- 2) граф имеет не более двух вершин нечётной степени.

Теорема (Эйлера). Конечный неориентированный граф $G = \langle V, E \rangle$ является эйлеровым тогда и только тогда, когда:

- 1) граф связан;
- 2) все степени его вершин чётны.

В задаче о мостах степени вершин графа нечётны, поэтому задача не имеет положительного решения.

Рассмотрим *алгоритм построения эйлерова цикла* в связном графе, все вершины которого имеют чётную степень.

- 1) составим списки инцидентности для данного графа;
- 2) выберем произвольно некоторую вершину a ;
- 3) выберем произвольное ребро, инцидентное данной вершине, и присвоим ему номер 1. Такое ребро назовём *пройденным*;
- 4) каждое пройденное ребро вычеркиваем и присваиваем ему номер, на единицу больший номера предыдущего вычеркнутого ребра;
- 5) после того, как все рёбра графа окажутся перенумерованными, получаем эйлеров цикл.

Эйлеровы пути и циклы известны с глубокой древности из головоломок, где нужно было нарисовать некоторую геометрическую фигуру, не отрывая руки и проходя через каждую вершину только один раз.

Рассмотрим некоторые задачи на эйлеровых графах.

Задача китайского почтальона. Эта задача состоит в том, чтобы пройти все улицы своего участка наиболее коротким путём и вернуться в начальную точку маршрута.

В терминах теории графов задача может быть сформулирована так. Строится граф, в котором ребро соответствует улице в маршруте почтальона, а каждая вершина – стыку двух улиц. Рёбрам графа приписаны некоторые положительные числа, называемые весами (например, длина улицы и т.п.). Требуется найти такой цикл, проходящий через каждое ребро графа, по крайней мере один раз, и такой, чтобы для него общий вес был бы наименьшим.

Задача уборки улиц города. Она состоит в следующем: необходимо составить маршрут движения уборочной машины по улицам заданного района города таким образом, чтобы он начинался и заканчивался в одном и том же месте и

проходил по каждой улице не более одного раза, если это возможно. Схему улиц района можно представить в виде графа, ребрам которого соответствуют улицы, а вершинам – перекрестки. В этом случае сформулированная задача сводится к построению эйлера цикла в графе.

Задача сбора мусора. Пусть определённый район города обслуживается единственной машиной-мусоровозом. Рёбра графа – это дороги, вершины – перекрёстки. Вес ребра равен длине дороги. Требуется найти цикл с минимальным километражем. Однако ёмкость мусоровоза и продолжительность рабочего дня накладывают ограничения на количество улиц, которые может обслужить одна машина. Поэтому может потребоваться нахождение не одного цикла, а некоторого конечного числа циклов, проходимых по одному в день, за период в несколько дней.

Другие приложения могут быть связаны с разбрасыванием смеси песка и соли при борьбе с гололёдом зимой на дорогах, с уборкой номеров и коридоров в гостиничных комплексах и больших учреждениях, с выбором наилучшего маршрута осмотра музея, с проблемами инспектирования распределённых систем и проверкой всех компонентов этих систем (например, электрических, телефонных и железнодорожных линий), с вопросами доставки населению питьевой вода или пиццы и т.д.

Задача 15.1 (старинная арабская головоломка). Можно ли, не отрывая руки, нарисовать фигуру, называемую «сабли Магомета»? (рис. 15.1).

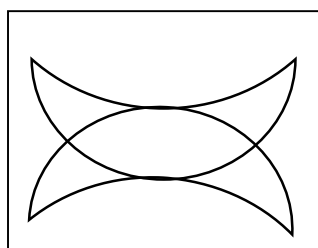


Рис. 15.1

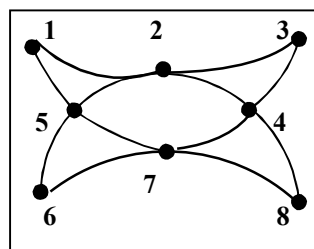


Рис. 15.2

Решение. Прежде всего попытаемся разглядеть граф в этом старинном рисунке. Для этого перенумеруем вершины (рис. 16.2). Покажем, что этот граф эйлеров. Степени всех вершин графа чётны: $d(1) = d(3) = d(6) = d(8) = 2$; $d(2) = d(4) = d(5) = d(7) = 4$. Согласно теореме (Эйлера) в графе существует эйлеров

цикл. Одно из решений задачи – цикл: 1-2-3-4-7-5-2-8-7-6-5-1; второе: 1-2-4-8-7-6-5-2-3-4-7-5-1.

Задача 15.2. Найти эйлеров цикл в графе (рис. 15.3).

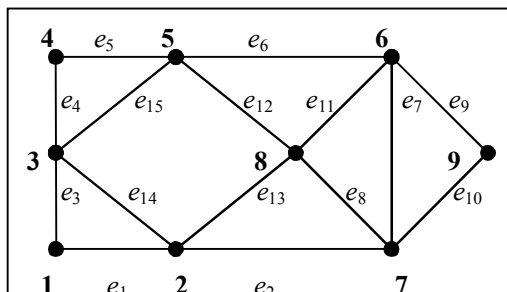


Рис. 15.3

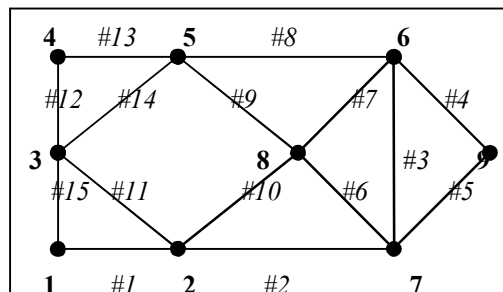


Рис. 15.4

Решение. Данный граф является эйлеровым. Составим списки вершин и инцидентных им рёбер данного графа (табл. 15.1).

Таблица 15.1

| <i>Вершина</i> | <i>Рёбра</i> |
|----------------|-------------------------------|
| 1 | e_1, e_3 |
| 2 | e_1, e_2, e_{13}, e_{14} |
| 3 | e_3, e_4, e_{14}, e_{15} |
| 4 | e_4, e_5 |
| 5 | e_5, e_6, e_{12}, e_{15} |
| 6 | e_6, e_7, e_9, e_{11} |
| 7 | e_2, e_7, e_8, e_{10} |
| 8 | $e_8, e_{11}, e_{12}, e_{13}$ |
| 9 | e_9, e_{10} |

Выбираем произвольную вершину (например **1**) и инцидентное ей ребро (например e_1). Присваиваем этому ребру номер 1 ($\#1$) и считаем его пройденным. Вычёркиваем ребро e_1 из списка вершин и рёбер. Переходим по e_1 в вершину **2**, выбираем её инцидентное ребро e_2 и присваиваем ему номер 2 ($\#2$). Вычёркиваем e_2 из списка и переходим в вершину **3**. Этот процесс продолжаем до тех пор, пока все рёбра не будут перенумерованы. Таким образом можно получить следующий эйлеров цикл (рис. 15.4):

1 -($\#1$)- **2**-($\#2$)- **7**- ($\#3$)- **6**- ($\#4$)- **9**- ($\#5$)- **7**- ($\#6$)- **8**- ($\#7$)- **6**- ($\#8$)- **5**- ($\#9$)- **8**-($\#10$)- **2**- ($\#11$)- **3**- ($\#12$)- **4**- ($\#13$)- **5**- ($\#14$)- **3**- ($\#15$)- **1**.

Задачи для самостоятельного решения.

1. Можно ли, не отрывая руки, нарисовать следующие фигуры (рис. 15.5)? Почему?

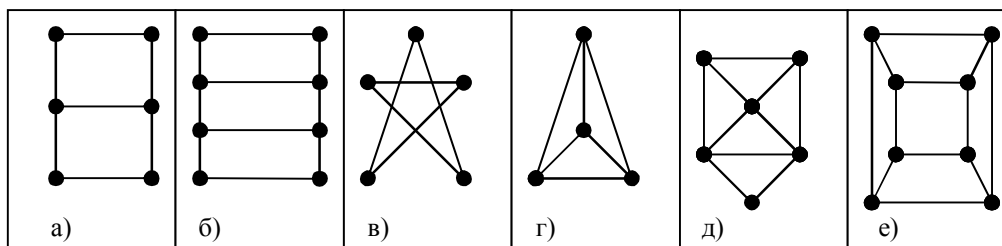


Рис. 15.5

(Ответ. Можно – а, в, д; нельзя – б, г, е).

2. Найти эйлеров цикл в графе G (рис. 15.6). (Ответ: 1-2-4-3-2-6-5-4-6-1).

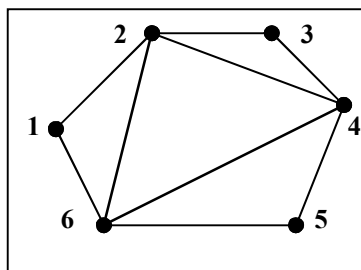


Рис. 15.6

16. ГАМИЛЬТОНОВЫ ГРАФЫ

В 1857 г. ирландский математик Гамильтон предложил игру, названную "Кругосветное путешествие". В ней использовался додекаэдр, каждой из двадцати вершин которого приписывалось название известного города мира. Каждый играющий должен был обойти «вокруг света» по замкнутому пути, который проходил бы через каждую вершину ровно один раз.

Додекаэдр - это многогранник, гранями которого служат 12 правильных пятиугольников. У него 20 вершин и 30 ребер. Граф, предложенный Гамильтоном, представляет собой укладку додекаэдра (рис. 16.1).

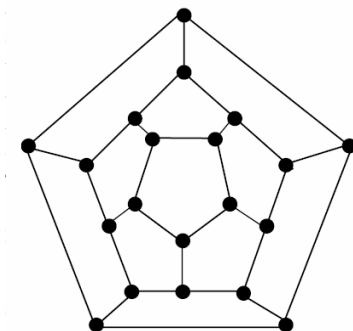


Рис. 16.1

В теории графов эта игра эквивалентна определению основного цикла, который содержит 20 вершин графа.

Гамильтонов цикл – это простой цикл, проходящий через все вершины графа. Заметим, что гамильтонов цикл есть далеко не в каждом графе. Граф называется *гамильтоновым*, если он содержит гамильтонов цикл.

Очевидно, что всякий полный граф является гамильтоновым. Кроме того, все правильные фигуры содержат такой цикл. К числу *правильных фигур* относят так называемые Платоновы тела или правильные многоугольники, такие как тетраэдр, куб, октаэдр, додекаэдр и икосаэдр (рис. 16.2).

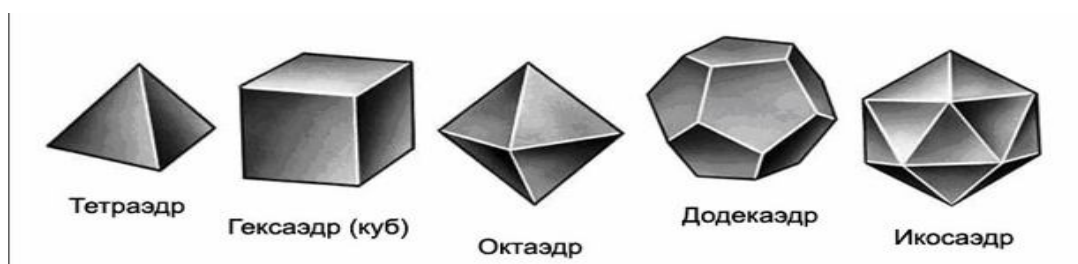


Рис. 16.2

На рис. 16.3 представлены графы соответствующих Платоновых тел (их укладка на плоскости).

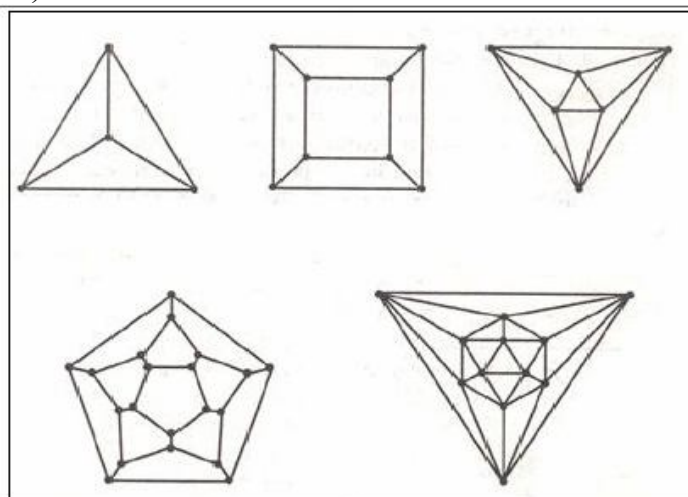


Рис. 16.3

Однако далеко не все графы имеют гамильтонов цикл. На рис. 16.4 представлен граф, не имеющий такого цикла. Это пример негамильтонового графа.

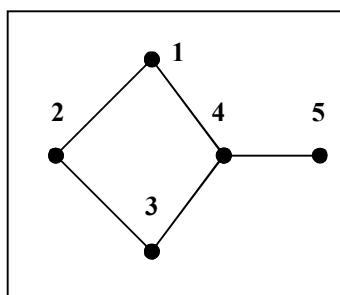


Рис. 16.4

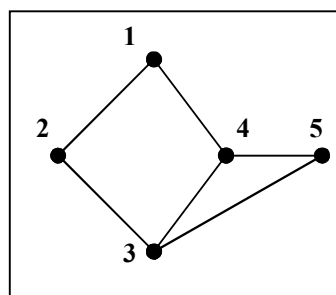


Рис. 16.5

Граф, изображенный на рис. 16.5, напротив, является гамильтоновым. Соответствующий цикл можно представить последовательностью рёбер: $\{(1,4), (4,5), (5,3), (3,2), (2,1)\}$. Можно заметить, что ребро $(4,3)$ в этом цикле не используется. Этот факт указывает на то, что в гамильтоновом цикле могут участвовать не все рёбра графа. Если граф не содержит кратных рёбер, то гамильтонов цикл в нём можно представить последовательностью вершин. Для графа на рис. 16.5 это будет последовательность $(1, 4, 5, 3, 2)$. Очевидно, что при такой записи каждый гамильтонов цикл должен соответствовать некоторой допустимой перестановке всех вершин графа. Ясно, что одному и тому же графу можно поставить в соответствие некоторое множество гамильтоновых циклов.

Задача о нахождении гамильтоновых циклов в графе является гораздо более сложной, чем аналогичная задача для эйлерового цикла, хотя эйлеровы и гамильтоновы графы сходны по способу задания. Первые содержат все рёбра по одному разу, вторые – все вершины тоже по одному разу.

Пока не найдено никакого простого критерия или алгебраического метода, позволяющего ответить на вопрос, существует или нет в произвольном графе G гамильтонов цикл. Известные критерии существования представляют теоретический интерес, но являются слишком общими и не пригодны для произвольных графов, встречающихся на практике. Большинство же известных фактов имеет вид: «если граф G имеет достаточное количество ребер, то граф является гамильтоновым».

Теорема. Если неориентированный граф G содержит гамильтонов цикл, тогда в нём не существует ни одной вершины v_i с локальной степенью $d(v_i) < 2$.

Данная теорема даёт необходимые условия «гамильтоновости» графа. Следующие же теоремы предоставляют достаточные условия существования этого факта.

Теорема Дирака (1952 г.). Пусть n – число вершин в простом графе G . Тогда, если $n \geq 3$ и степень каждой вершины не меньше, чем $n/2$, то граф G – гамильтонов.

Теорема Оре (1960 г.) Пусть n – количество вершин в графе G . Если для любой пары несмежных вершин x и y выполнено неравенство $d(x) + d(y) \geq n$ (то есть степени любых двух несмежных вершин не меньше общего числа вершин в графе), то граф G – гамильтонов.

Задачи для самостоятельного решения.

1. Найти гамильтоновы циклы на графах G_1 , G_2 и G_3 (рис. 16.6).
 (ответ для G_3 : 10-6-7-5-4-3-2-8-9-1-10).

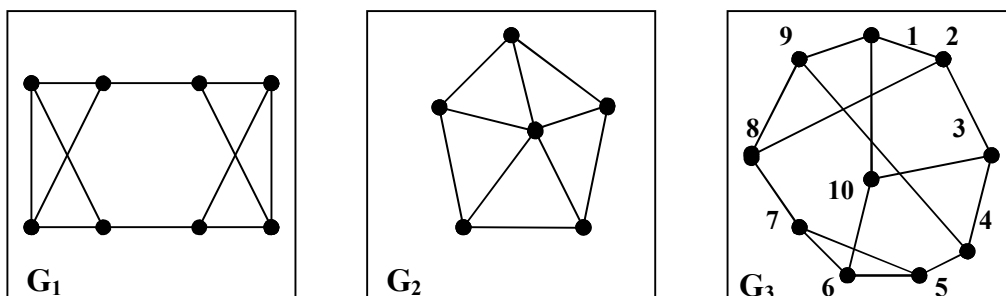


Рис. 16.6

17. НЕКОТОРЫЕ ЗАДАЧИ НА ГАМИЛЬТОНОВЫХ ГРАФАХ

Задача коммивояжера. Задача коммивояжера является одной из знаменитых задач теории комбинаторики. Она была поставлена в 1934 году, и об неё, как и о Великую теорему Ферма, «обламывали зубы» многие математики. В своей области (оптимизации дискретных задач) эта задача служит своеобразным полигоном, на котором испытываются всё новые методы.

Постановка задачи такова. На территории некоторого государства расположено n городов, между некоторыми парами которых проложены дороги (не обязательно двунаправленные). Каждая из них характеризуется целым положительным числом – своей длиной. Бродячий торговец хочет обойти все города ровно по одному разу и вернуться в исходный пункт (начинать движение можно в произвольном городе). Необходимо найти такой способ обхода, при котором суммарная длина пройденных дорог была бы минимальной. Или определить, что искомого маршрута не существует.

В терминах теории графов задачу можно переформулировать так: найти в заданном взвешенном ориентированном графе гамильтонов цикл минимального веса или выяснить, что его не существует.

Первая идея, которая приходит в голову при попытке решения данной задачи, это перебрать все возможные пути, проходящие только по различным вершинам, и для каждого проверить соответствие определению гамильтонова пути (цикл получится из пути добавлением одного ребра). Асимптотическая сложность такого алгоритма состоит в том, что такой алгоритм будет работать неприемлемо долго даже на небольших графах.

Задача коммивояжёра относится к так называемому классу NP. К настоящему времени не найдено полиномиальных алгоритмов, позволяющих получить точное решение этой задачи. Самый эффективный из известных алгоритмов для поиска решения имеет асимптотическую сложность и основан на динамическом программировании.

Задача о шахматном коне. Можно ли шахматным конём, начиная с произвольной клетки шахматной доски, обойти все 64 клетки и вернуться в исходную точку, побывав в каждой клетке ровно один раз?

Эта задача интересовала многих математиков, особенно Эйлера, де Муавра, Вандермонда и др. Правило, которое, видимо, оправдывается на практике, но еще не подтверждено теоретически, состоит в следующем: всякий раз ходим конем туда, где находится наименьшее число еще не пройденных клеток.

Другой способ состоит в нахождении маршрута по половине доски, а затем симметричном дублировании его и соединении обоих маршрутов.

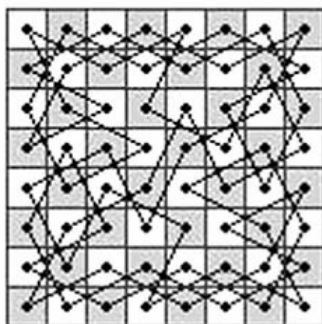


Рис. 17.1

В следующей таблице клетки шахматной доски занумерованы в порядке обхода конём сначала половины доски и дальнейшим симметричным продолжением на другую половину.

Таблица 17.1

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 30 | 3 | 22 | 9 | 28 | 5 | 24 | 15 |
| 21 | 10 | 29 | 4 | 23 | 16 | 27 | 6 |
| 2 | 31 | 12 | 19 | 8 | 25 | 14 | 17 |
| 11 | 20 | 1 | 32 | 13 | 18 | 7 | 26 |
| 58 | 39 | 50 | 45 | 64 | 33 | 52 | 43 |
| 49 | 46 | 57 | 40 | 51 | 44 | 63 | 34 |
| 38 | 59 | 48 | 55 | 36 | 61 | 42 | 53 |
| 56 | 56 | 37 | 60 | 41 | 54 | 35 | 62 |

Задача о банкете (о званом обеде). Обед накрыт на круглом столе. Некоторые гости знакомы друг с другом. Удастся ли хозяевам рассадить приглашенных так, чтобы по обе стороны от каждого находились его знакомые?

Решение задачи задаёт гамильтонов цикл на графе знакомств компании.

До настоящего времени не найдены эффективные алгоритмы решения подобных задач. Если попытаться решить их простым перебором вершин графа, то это можно сделать лишь для графов с небольшим количеством вершин, так как они требуют слишком большого времени работы и большой памяти компьютера.

Так для графа с n вершинами необходимо вычислить $n!$ перестановок и для каждой провести анализ, не задаёт ли она гамильтонов цикл. Для реализации этого анализа потребуется $n \cdot n!$ вычислительных операций (то есть n операций на каждую из $n!$ перестановок).

Например, для графа с $n = 20$ вершинами для поиска гамильтонова цикла потребуется $20 \cdot 20! = 48 \cdot 10^{18}$ операций. Если положить, что каждый элементарный шаг компьютер выполняет за время, равное 10^{-7} секунд, то для решения данной задачи потребуется ни много, ни мало – 1500 столетий.

Те задачи, время решения которых (количество операций для нахождения ответа) зависит от количества входных данных как многочлен, называются принадлежащими к Р-классу (polynomial) алгоритмов. Те задачи, для которых можно хотя бы проверить правильность предложенного решения за полиномиально зависящее от входных данных время, называются принадлежащими NP-классу (non-deterministic polynomial). Р-класс входит в NP-класс. А NP-полные задачи - те, для которых не доказано ни то, что они не могут быть решены за полиномиальное время, ни то, что они не могут быть

решены за полиномиальное время (то есть они NP, но неизвестно, может, они и P, но это невозможно доказать).

18. ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ НА ГРАФАХ

Постановка многих прикладных задач может быть сформулирована в точках и связях между ними, то есть в терминах теории графов. Так можно сформулировать задачи составления расписаний, анализа сетей в электротехнике, анализа цепей Маркова в теории вероятностей, в программировании, в экономике, социологии и других науках. Поэтому эффективные алгоритмы решений теории графов имеют большое практическое значение.

Все имеющиеся задачи можно условно разделить на следующие классы.

Задача выбора кратчайшего пути. Это задача выбора оптимального маршрута доставки продукции от поставщиков к потребителям с учетом затрат на перевозки по имеющимся участкам транспортной сети. Отметим, что к этой задаче можно свести многие оптимизационные задачи, возникающие в финансах и экономике, например, задачу о поиске наилучшей стратегии размещения капитала.

Задача о максимальном потоке в сети. В транспортных сетях (в том числе нефтепроводах, системах передачи информации и т. д.) могут использоваться элементы (как каналы, так и узлы системы) с различной пропускной способностью. Поэтому, естественно, возникает задача определения максимального потока, который можно передать от определенного отправителя к конкретному получателю, с учетом пропускных способностей каналов и узлов системы.

Задачи сетевого планирования. Цель сетевого планирования – воздействие на управление, а управление призвано поддерживать рациональный режим работы, восстанавливать нарушенное состояние подвижного равновесия динамических систем, обеспечивая слаженную работу всех ее звеньев. При этом управление системой ведется по ряду параметров: времени, стоимости, ресурсам, технико-экономическим показателям. Однако наиболее распространенными являются системы с параметром «время».

Основу сетевой модели составляет сетевой график – наглядное представление плана работ в виде информационной модели, отображающей

процесс выполнения комплекса работ, направленных на достижение единой цели. Он позволяет руководству программы точно знать, что требуется делать в каждый момент времени и кто именно должен это делать, а также вероятность своевременного завершения отдельных операций.

При решении этих задач на графах характерной особенностью является то, что рёбрам графа приписываются некоторые числа, которые называют весом ребра. Такой граф называется *взвешенным* (или *нагруженным*) графом. Часто такой граф называют также *сетью*.

18. 1. ПОСТРОЕНИЕ СЕТИ НАИМЕНЬШЕЙ ДЛИНЫ. АЛГОРИТМ КРАСКАЛА

Подобные задачи возникают в фирмах, занимающихся прокладками коммуникационных сетей для компьютерных систем, систем кабельного телевидения, энергетических и телефонных сетей и т.д.

Задача. Телевизионная фирма планирует создание кабельной сети для обслуживания некоторого района города. Совершенно очевидно, что общая длина прокладываемого кабеля должна иметь минимальную длину. Если рассматривать здания района как вершины графа, а кабели между зданиями как ребра, то эта работа с прокладыванием кабеля превращается в задачу определения сети, охватывающей все здания территории района, в которой общая длина проложенного кабеля должна быть минимальной.

Такую задачу называют нахождением сети минимального веса (длины), задачей о минимальном соединении, задачей о нахождении минимального остовного дерева (или просто минимального остова, или минимального каркаса) графа.

Минимальным остовным деревом связного взвешенного графа называется его связный подграф, состоящий из всех вершин исходного графа и некоторых его ребер, причем сумма весов ребер минимально возможная. Если исходный граф несвязен, то описываемую ниже процедуру можно применять поочередно к каждой его компоненте связности, получая тем самым минимальные остовные деревья для этих компонент.

Минимальный остов уникален, если веса всех рёбер различны. В противном случае, может существовать несколько минимальных остовов (конкретные алгоритмы обычно получают один из возможных остовов).

Существует несколько алгоритмов для нахождения минимального остовного дерева. Наиболее известные из них: алгоритм Прима, алгоритм Краскала (или алгоритм Крускала) и алгоритм Борувки.

Рассмотрим алгоритм, достаточно привлекательный в вычислительном отношении, который был предложен Джозефом Краскалом в 1957 году.

Пусть задан взвешенный неориентированный граф $G = \langle V, E \rangle$ с n вершинами и m рёбрами. Требуется найти такое дерево этого графа, которое бы соединяло все его вершины, и при этом обладало наименьшим возможным весом (т.е. суммой весов рёбер).

Алгоритм Краскала состоит из двух фаз:

1. На подготовительной (первой) фазе все рёбра удаляются из графа и упорядочиваются по возрастанию их весов. В графе остаются только вершины, и каждая из них образует отдельную компоненту связности.

2. Во второй фазе рёбра перебираются в порядке возрастания веса. Если начало и конец очередного ребра принадлежат одной и той же компоненте связности, ребро игнорируется. Если же они лежат в разных компонентах связности, ребро добавляется к графу, а эти две компоненты связности объединяются в одну. Если число компонент связности дойдет до 1, цикл завершается досрочно.

Задача 18.1.1. Рассмотрим работу алгоритма на примере взвешенного графа, заданного матрицей весов Ω :

$$\Omega = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{pmatrix} - & 4 & \infty & 3 & 3 & 1 \\ 4 & - & 4 & \infty & 2 & 5 \\ \infty & 4 & - & 6 & \infty & 3 \\ 3 & \infty & 6 & - & 4 & 5 \\ 3 & 2 & \infty & 4 & - & \infty \\ 1 & 5 & 3 & 5 & \infty & - \end{pmatrix} \end{matrix}$$

Вес каждого ребра записан на пересечении соответствующих строк и столбцов. Если же вершины не соединены ребром, то на пересечении ставится символ ∞ . Поскольку граф неориентированный, матрица его весов симметрична относительно своей главной диагонали. Граф, заданный матрицей Ω , изображен на рис. 18.1.

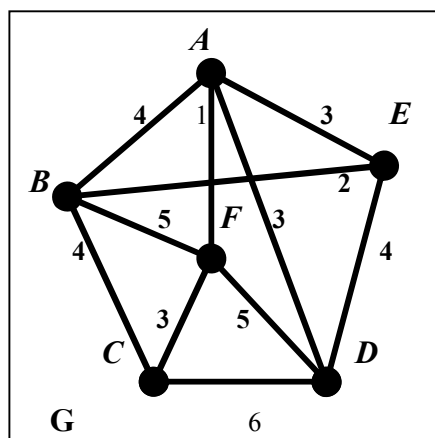


Рис. 18.1

Решение.

Подготовительная фаза. Сначала составим список рёбер графа G. Результаты занесём в таблицу 18.1.

Таблица 18.1

| Вершина 1 | Вершина 2 | Вес ребра |
|-----------|-----------|-----------|
| <i>A</i> | <i>B</i> | 4 |
| <i>A</i> | <i>D</i> | 3 |
| <i>A</i> | <i>E</i> | 3 |
| <i>A</i> | <i>F</i> | 1 |
| <i>B</i> | <i>C</i> | 4 |
| <i>B</i> | <i>E</i> | 2 |
| <i>B</i> | <i>F</i> | 5 |
| <i>C</i> | <i>D</i> | 6 |
| <i>C</i> | <i>F</i> | 3 |
| <i>D</i> | <i>E</i> | 4 |
| <i>D</i> | <i>F</i> | 5 |

Произведём сортировку рёбер графа G и расположим их в порядке возрастания весов (табл. 18.2).

Таблица 18.2.

| Вершина 1 | Вершина 2 | Вес ребра |
|-----------|-----------|-----------|
| <i>A</i> | <i>F</i> | 1 |
| <i>B</i> | <i>E</i> | 2 |
| <i>A</i> | <i>D</i> | 3 |
| <i>A</i> | <i>E</i> | 3 |
| <i>C</i> | <i>F</i> | 3 |
| <i>A</i> | <i>B</i> | 4 |
| <i>B</i> | <i>C</i> | 4 |
| <i>D</i> | <i>E</i> | 4 |
| <i>B</i> | <i>F</i> | 5 |
| <i>D</i> | <i>F</i> | 5 |
| <i>C</i> | <i>D</i> | 6 |

Теперь построим пустой граф $T = \langle V, \emptyset \rangle$, который имеет то же множество вершин V , что и исходный, но не имеет рёбер. Он представлен на рис. 18.2.

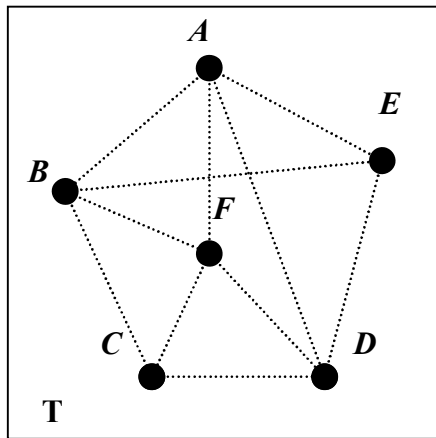


Рис. 18.2

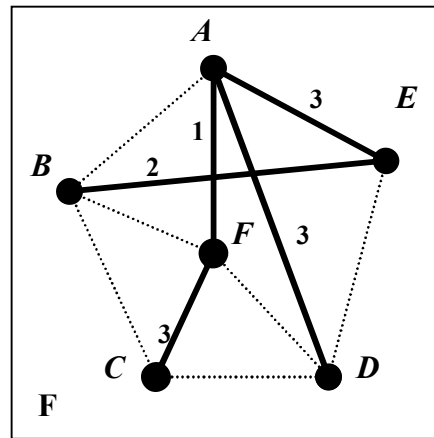


Рис. 18.3

На этом заканчивается подготовительная фаза алгоритма Краскала.

Вторая фаза.

Шаг 1. Первое ребро, которое будет рассмотрено – (A, F) , поскольку его вес минимален. Добавим это ребро к остовному дереву, которое будет расти по мере перебора рёбер. Теперь имеем новую компоненту связности – две вершины $(A$ и $F)$ и инцидентное им ребро (табл.18.3).

Шаг 2. Ребро (B, E) добавляем, поскольку оно не принадлежит полученной на шаге 1 компоненте связности.

Шаг 3. Ребро (A, D) добавляем, поскольку оно не принадлежит найденным ранее компонентам связности.

Шаг 4. Ребро (A, E) добавляется, так как оно принадлежит различным компонентам связности.

Шаг 5. Ребро (C, F) добавляется по тем же причинам. Полученная на этом шаге компонента связности охватывает все вершины, и добавление любого нового ребра приведёт к образованию цикла, то есть построенный граф не будет являться деревом. Поэтому процесс построения минимального остова графа G завершен.

Ответ задачи представлен на рис. 18.3. Это и есть минимальный остовный граф $F = \langle V, E_F \rangle$ исходного графа $G = \langle V, E \rangle$. Граф F имеет то же множество вершин V , что и граф G , а множество рёбер F включено в множество рёбер графа G : $(E_F \subset E)$.

Длина пути минимального остовного графа равна сумме длин всех входящих в него рёбер:

$$l = l_{AF} + l_{BE} + l_{AD} + l_{AE} + l_{CF} = 1 + 2 + 3 + 3 + 3 = 12 \text{ (усл. ед.)}.$$

Таблица 18.3.

| № шага | Вершина 1 | Вершина 2 | Вес ребра | Признак добавления ребра | Растущее минимальное остовное дерево |
|--------|-----------|-----------|-----------|--------------------------|--------------------------------------|
| 1 | <i>A</i> | <i>F</i> | 1 | + | |
| 2 | <i>B</i> | <i>E</i> | 2 | + | |
| 3 | <i>A</i> | <i>D</i> | 3 | + | |
| 4 | <i>A</i> | <i>E</i> | 3 | + | |
| 5 | <i>C</i> | <i>F</i> | 3 | + | |
| 6 | <i>A</i> | <i>B</i> | 4 | - | - |
| 7 | <i>B</i> | <i>C</i> | 4 | - | - |
| 8 | <i>D</i> | <i>E</i> | 4 | - | - |
| 9 | <i>B</i> | <i>F</i> | 5 | - | - |
| 10 | <i>D</i> | <i>F</i> | 5 | - | - |
| 11 | <i>C</i> | <i>D</i> | 6 | - | - |

Задачи для самостоятельного решения.

1. По заданной матрице весов Ω графа G найти сеть минимальной длины этого графа с помощью алгоритма Краскала.

$$\text{a) } \begin{pmatrix} - & 10 & 5 & 2 & 16 & \infty & \infty \\ 10 & - & 4 & \infty & \infty & \infty & \infty \\ 5 & 4 & - & 8 & 15 & 13 & \infty \\ 2 & \infty & 8 & - & 8 & 5 & 7 \\ 16 & \infty & 15 & 8 & - & 11 & 18 \\ \infty & \infty & 13 & 5 & 11 & - & 4 \\ \infty & \infty & \infty & 7 & 18 & 4 & - \end{pmatrix}; \quad \text{b) } \begin{pmatrix} - & 2 & \infty & 7 & \infty & \infty & \infty \\ 2 & - & 1 & 8 & 4 & \infty & 10 \\ \infty & 1 & - & \infty & 5 & 6 & 9 \\ 7 & 8 & \infty & - & \infty & 3 & \infty \\ \infty & 4 & 5 & \infty & - & 4 & 10 \\ \infty & \infty & 6 & 3 & 4 & - & 4 \\ \infty & 10 & 9 & \infty & 10 & 4 & - \end{pmatrix}.$$

18.2. ПОИСК КРАТЧАЙШИХ ПУТЕЙ ИЗ ДАННОЙ ВЕРШИНЫ ГРАФА КО ВСЕМ_ОСТАЛЬНЫМ

Задача 1. Дана сеть автомобильных дорог, соединяющих города области. Некоторые дороги односторонние. Найти кратчайшие пути от областного центра до каждого города области (если двигаться можно только по дорогам).

Задача 2. Имеется некоторое количество авиарейсов между городами мира, для каждого известна стоимость. При этом стоимость перелёта из А в В может быть не равна стоимости перелёта из В в А. Найти маршрут минимальной стоимости (возможно, с пересадками) от Копенгагена до Днепропетровска.

Существует несколько методов поиска кратчайших путей в графе: алгоритм Дейкстры (алгоритм поиска кратчайших путей от заданной вершины), алгоритм Флойда (алгоритм поиска длин всех кратчайших путей в графе), алгоритм Данцига.

Рассмотрим *алгоритм Дейкстры*, предложенный в 1959 г., который считается одним из наиболее эффективных алгоритмов решения задачи.

Пусть $G = \langle V, E \rangle$ – связный взвешенный орграф без петель и дуг отрицательного веса. Каждой дуге (x, y) этого графа поставим в соответствие число $a(x, y)$. Если в графе отсутствует некоторая дуга (x, y) , положим $a(x, y) = \infty$. Будем называть число $a(x, y)$ длиной дуги (x, y) , хотя $a(x, y)$ можно также интерпретировать как соответствующие затраты или соответствующий весовой коэффициент. Определим длину пути как сумму длин отдельных дуг, составляющих этот путь.

Для любых двух вершин s и t графа G могут существовать несколько путей, соединяющих вершину s с вершиной t . В теории графов рассматривается задача, цель которой определить такой путь, ведущий из вершины s в вершину t , который имеет минимально возможную длину. Этот путь называется кратчайшим путем между вершинами s и t .

Главная идея, лежащая в основе алгоритма Дейкстры, предельно проста. Предположим, что нам известны m вершин, ближайших к вершине s (близость любой вершины x к вершине s определяется длиной кратчайшего пути, ведущего из s в x).

Пусть также известны сами кратчайшие пути, соединяющие вершину s с выделенными m вершинами. Покажем теперь, как может быть определена $(m + 1)$ -я ближайшая к s вершина.

Окрасим (или выделим каким-либо образом) вершину s и m ближайших к ней вершин. Построим для каждой неокрашенной вершины y пути, непосредственно соединяющие с помощью дуг (x, y) каждую окрашенную вершину x с y . Выберем из этих путей кратчайший, и будем считать его условно кратчайшим путем из вершины s в вершину y .

Какая же из неокрашенных вершин является $(m + 1)$ -й ближайшей к s вершиной? Та, для которой условно кратчайший путь имеет наименьшую длину. Это обуславливается тем, что кратчайший путь из вершины s в $(m + 1)$ -ю ближайшую вершину при положительном значении длин всех дуг должен содержать в качестве промежуточных лишь окрашенные вершины, т. е. вершины, входящие в число m вершин, ближайших к вершине s .

Итак, если известны m ближайших к s вершин, то $(m + 1)$ -я ближайшая к s вершина может быть найдена так, как это описано выше. Начиная с $m = 0$, описанная процедура может повторяться до тех пор, пока не будет получен кратчайший путь, ведущий из вершины s к вершине t .

Формальное описание алгоритма Дейкстры.

Шаг 1. Каждой вершине x в ходе алгоритма присваивается число $d(x)$, равное длине кратчайшего пути из вершины s в вершину x и включающем только окрашенные вершины. Положим $d(s) = 0$ и $d(x) = \infty$ для всех остальных вершин графа. Окрашиваем вершину s и полагаем $y = s$, где y – последняя окрашенная вершина.

Для каждой неокрашенной вершины x пересчитывается величина $d(x)$ по следующей формуле:

$$d(x) = \min \{d(x); d(y) + a(y, x)\} \quad (*)$$

Шаг 2. Если $d(x) = \infty$ для всех неокрашенных вершин, то алгоритм заканчивается, т. к. отсутствуют пути из вершины s в неокрашенные вершины. Иначе, окрашивается та вершина, для которой величина $d(x)$ является минимальной. Окрашивается и дуга, ведущая в эту вершину в соответствии с выражением (*). Далее полагаем $y = x$.

Таким образом, каждый раз окрашивается вершина и дуга, заходящая в эту вершину. Окрашенные дуги не могут образовывать цикл, а образуют в исходном графе дерево с корнем (началом) в вершине s . Это дерево называют ориентированным деревом кратчайших путей. Путь из s в t принадлежит этому дереву. При поиске одного кратчайшего пути процедура наращивания

завершается при достижении конечной вершины этого пути. Нам же необходимо получить все кратчайшие пути, начинающиеся в вершине s . Для этого процедура наращивания ориентированного дерева продолжается до тех пор, пока все вершины не будут включены. Таким образом мы получаем ориентированное дерево кратчайших путей.

Иногда в графе имеются несколько кратчайших путей. Кратчайший путь будет единственным, если в алгоритме ни разу не возникает неоднозначность при окрашивании дуги.

Если $y = t$, кратчайший путь из s в t найден. Иначе возвращаемся к шагу 2.

Задача 18.2.1. Рассмотрим работу алгоритма Дейкстры на примере нагруженного орграфа с матрицей весов Ω :

$$\Omega = \begin{pmatrix} - & 10 & 28 & 8 & \infty & \infty \\ \infty & - & 16 & 9 & 21 & \infty \\ \infty & \infty & - & \infty & 9 & \infty \\ \infty & 9 & 12 & - & \infty & 22 \\ \infty & 10 & 20 & \infty & - & 23 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}$$

Для простоты обозначим вершины графа цифрами от 1 до 6. Необходимо найти все кратчайшие пути от вершины 1. Схема графа представлена на рисунке 18.4.

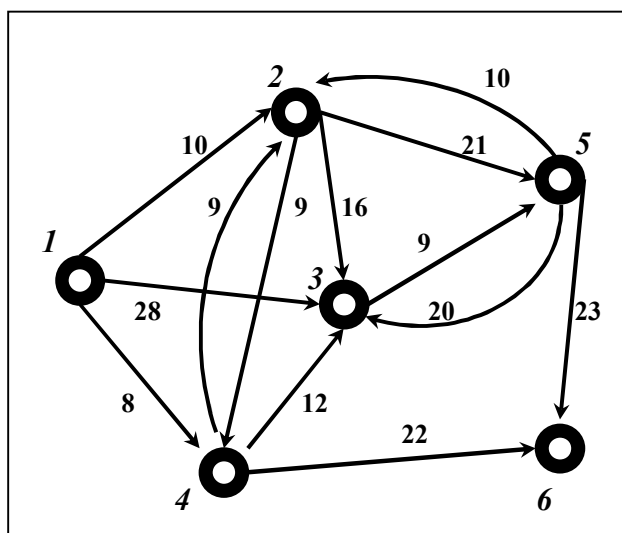


Рис. 18.4

Решение.

Шаг 1. Стартовая вершина, от которой строится дерево кратчайших путей вершина 1 (она же вершина s). Конечной считаем вершину 6 (она же вершина t).

Задаем стартовые условия: $d(1)=0$, $d(x) = \infty$, где $x = 2, 3, 4, 5, 6$.

Окрашиваем вершину 1 , полагаем $y = 1$ (y – это последняя окрашенная вершина).

Находим ближайшую вершину к окрашенной нами, используя формулу:

$$\underline{d(x) = \min \{d(x); d(y) + a(y, x)\}.$$

$$d(2) = \min \{d(2); d(1) + a(1, 2)\} = \min \{\infty; 0 + 10\} = 10,$$

$$d(3) = \min \{d(3); d(1) + a(1, 3)\} = \min \{\infty; 0 + 28\} = 28,$$

$$d(4) = \min \{d(4); d(1) + a(1, 4)\} = \min \{\infty; 0 + 8\} = 8,$$

$$d(5) = \min \{d(5); d(1) + a(1, 5)\} = \min \{\infty; 0 + \infty\} = \infty,$$

$$d(6) = \min \{d(6); d(1) + a(1, 6)\} = \min \{\infty; 0 + \infty\} = \infty.$$

Минимальную длину имеет путь от вершины 1 до вершины 4 : $d(4) = 8$. Включаем вершину 4 в текущее ориентированное дерево, а также дугу, ведущую в эту вершину. Согласно выражению для $d(4)$ - это дуга $(1, 4)$. Окрашиваем вершину 4 и дугу $(1, 4)$. На рисунке 18.5 представлено растущее дерево кратчайших путей от вершины 1 .

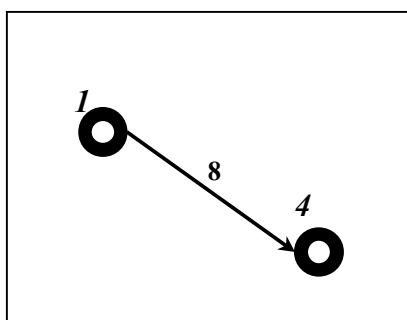


Рис. 18.5

Шаг 2.

Полагаем $y = 4$. Находим ближайшую вершину к окрашенной нами, используя формулу:

$$\underline{d(x) = \min \{d(x); d(y) + a(y, x)\}.$$

$$d(2) = \min \{d(2); d(4) + a(4, 2)\} = \min \{10; 8 + 9\} = 10,$$

$$d(3) = \min \{d(3); d(4) + a(4, 3)\} = \min \{28; 8 + 12\} = 20,$$

$$d(5) = \min \{d(5); d(4) + a(4, 5)\} = \min \{\infty; 8 + \infty\} = \infty,$$

$$d(6)=\min\{d(6) ; d(4)+a(4,6)\}=\min\{\infty; 8+22\}=30.$$

Минимальную длину имеет путь от вершины 1 до вершины 2 : $d(2) = 10$. Включаем вершину 2 в текущее ориентированное дерево, а также дугу, ведущую в эту вершину. Согласно выражению для $d(2)$ - это дуга $(1,2)$. Окрашиваем вершину 2 и дугу $(1,2)$.

На рисунке 18.6 изображено дерево наименьших путей.

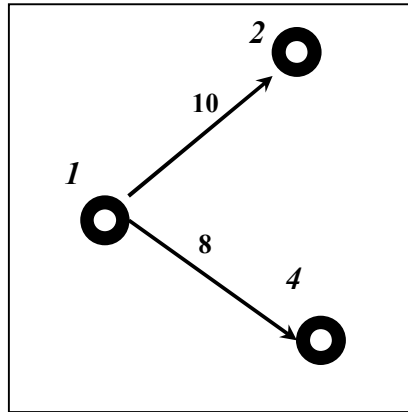


Рис. 18.6

Замечание. В вершину 2 из вершины 1 можно попасть и через окрашенную уже вершину 4 . Однако путь $1 - 4 - 2$ будет имеет длину $6 + 9 = 17$, а путь $1 - 2$ равен 10 . Поэтому мы и окрасили дугу $(1,2)$, а не $(4,2)$.

Поскольку вершина t (она же 6) ещё не достигнута, снова выполняем шаг 2.

Шаг 2*.

Полагаем $y = 2$. Находим ближайшую вершину к окрашенной нами, используя формулу:

$$\underline{d(x) = \min\{d(x); d(y)+ a(y, x)\}.$$

$$d(3)=\min\{d(3) ; d(2)+a(2,3)\}=\min\{20; 10+16\}=20,$$

$$d(5)=\min\{d(5) ; d(2)+a(2,5)\}=\min\{\infty; 10+21\}=31,$$

$$d(6)=\min\{d(6) ; d(2)+a(2,6)\}=\min\{30; 10+\infty\}=30.$$

Минимальную длину имеет путь от вершины 1 до вершины 3 : $d(3) = 20$. Включаем вершину 3 в текущее ориентированное дерево, а вот дугу $(1,3)$, ведущую в эту вершину, включать (окрашивать) в растущее дерево нельзя.

Дело в том, что путь из 1 в 3 по дуге $(1,3)$ будет равен 28, а по дугам $(1,4)$ и $(4,3)$ в сумме $8+12 = 20$. Поэтому в дерево включаем вершину 3 и дугу $(4,3)$. На рисунке 18.7 изображено растущее дерево наименьших путей.

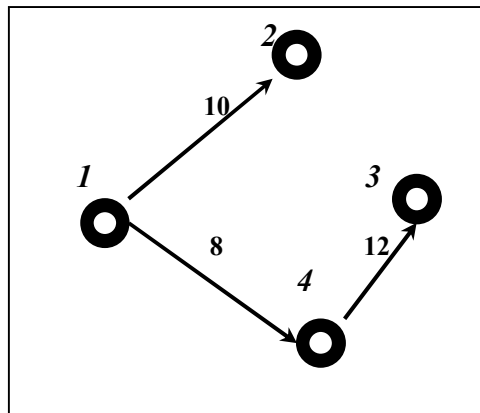


Рис. 18.7

Поскольку вершина t (она же 6) ещё не достигнута, снова выполняем шаг 2.

Шаг 2 **.

Полагаем $y = 3$. Находим ближайшую вершину к окрашенной нами, используя формулу:

$$\underline{d(x) = \min \{d(x); d(y) + a(y, x)\}}.$$

$$d(5) = \min \{d(5); d(3) + a(3, 5)\} = \min \{31; 20 + 9\} = 29,$$

$$d(6) = \min \{d(6); d(3) + a(3, 6)\} = \min \{30; 20 + \infty\} = 30.$$

Минимальную длину имеет путь от вершины 1 до вершины 5 : $d(5) = 29$. Включаем вершину 5 в растущее ориентированное дерево, а также дугу $(3,5)$ (а не дугу $(2,5)$!), ведущую в эту вершину. Растущее дерево наименьших путей представлено на рис. 18.8.

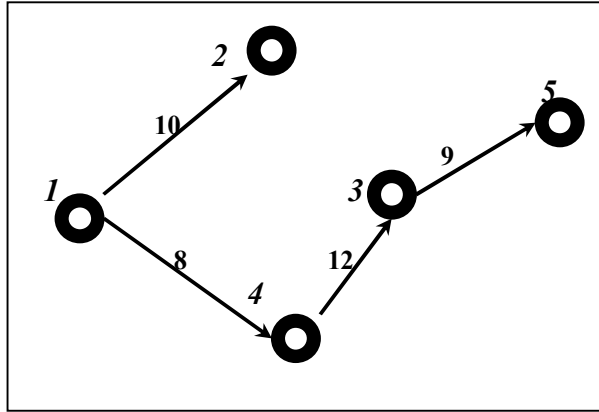


Рис. 18.8

Поскольку вершина t (она же 6) ещё не достигнута, снова выполняем шаг 2.

Шаг 2***.

Полагаем $y = 5$. Находим ближайшую вершину к окрашенной нами, используя формулу:

$$\underline{d(x) = \min \{d(x); d(y) + a(y, x)\}}.$$

$$d(6) = \min \{d(6); d(5) + a(5, 6)\} = \min \{30; 29 + 23\} = 30.$$

Минимальную длину имеет путь от вершины 1 до вершины 6 : $d(6) = 30$.

Включаем вершину 6 в растущее ориентированное дерево, а также дугу, ведущую в эту вершину из вершины 4 – дугу $(4, 6)$.

Конечная вершина t (она же 6) достигнута. Мы получили ориентированное дерево с корнем в вершине 1 исходного графа (рис. 18.9).

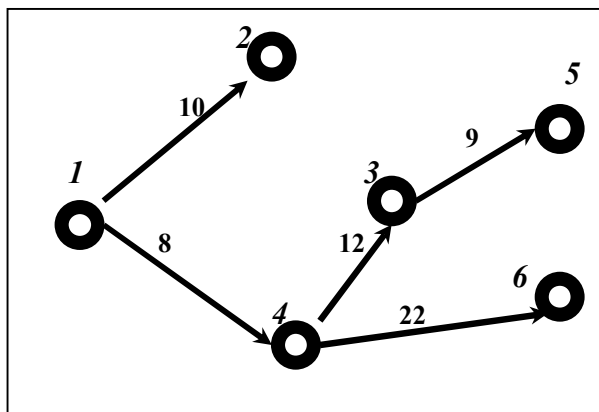


Рис. 18.9

$$d(1) = 0 \text{ - длина пути } 1-1;$$

$$d(2) = 10 \text{ - длина пути } 1-2;$$

$$d(3) = 20 \text{ - длина пути } 1-4-3;$$

$$d(4) = 8 \text{ - длина пути } 1-4;$$

$d(5) = 29$ - длина пути $1-4-3-5$;

$d(6) = 30$ - длина пути $1-4-6$.

Величину минимального пути от вершины s до вершины t даёт нам число $d(6)$. Этот путь пролегает через вершины $1-4-6$ и имеет длину $l = 8+22 = 30$ (усл. ед.).

Задачи для самостоятельного решения.

1. По заданной матрице весов Ω графа G найти величину минимального пути и сам путь от вершины s до вершины t .

$$a) \begin{pmatrix} - & 5 & 10 & 13 & \infty & \infty \\ \infty & - & 8 & 9 & 13 & \infty \\ \infty & \infty & - & 5 & 3 & 6 \\ \infty & \infty & \infty & - & 8 & 10 \\ \infty & \infty & \infty & \infty & - & 9 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}; \quad b) \begin{pmatrix} - & 11 & \infty & 14 & 15 & \infty \\ \infty & - & 13 & \infty & \infty & \infty \\ \infty & \infty & - & \infty & \infty & 13 \\ \infty & 7 & 11 & - & 9 & \infty \\ \infty & 11 & 10 & \infty & - & 14 \\ \infty & \infty & \infty & \infty & \infty & - \end{pmatrix}; \quad c) \begin{pmatrix} - & 5 & 8 & 7 & 18 & \infty \\ \infty & - & 11 & \infty & \infty & \infty \\ \infty & \infty & - & \infty & \infty & 17 \\ \infty & 10 & 12 & - & 6 & \infty \\ \infty & 7 & 8 & \infty & - & 11 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix}.$$

18.3. НАХОЖДЕНИЕ МАКСИМАЛЬНОГО ПОТОКА В СЕТИ

Рассмотрим взвешенный оргграф $G = \langle V, E \rangle$ с двумя выделенными вершинами – *источником* (s) и *стоком* (t). Источник имеет нулевую полустепень захода (то есть ни одна дуга не входит в эту вершину), сток – нулевую полустепень исхода (ни одна дуга не выходит из этой вершины). Каждой дуге приписывается два целых неотрицательных числа, записываемых в виде дроби $c(i, j) / \varphi(i, j)$. Вес дуги $c(i, j)$ – число в числителе – означает её пропускную способность и показывает, сколько вещества может быть пропущено из вершины i в вершину j (является числом постоянным). Поток $\varphi(i, j)$ – это ещё одно число, приписанное дуге, которое записывается в знаменателе и показывает, сколько вещества движется в данный момент из вершины i в вершину j . Поток дуги не больше её пропускной способности ($0 \leq \varphi(i, j) \leq c(i, j)$) и может меняться.

Поток выходит из источника и без потерь в том же объёме заходит в сток. Условие равновесия (по объёму входа и выхода) выполняется и для каждой вершины сети.

Определим величину потока $f = \sum_{i \in V} \varphi(s, i)$ как сумму по всем вершинам i , т.е. суммарный поток по всем ребрам, выходящим из источника s .

Задача о наибольшем потоке в сети – не единственная, но, вероятно, основная задача для потоков в сетях. Практическое применение этой задачи может быть найдено в решении транспортных проблем на дорогах (например, пробки на дорогах можно трактовать как насыщение сети или отдельной её дуги), проблем транспортировки нефтепродуктов, газа, электроэнергии и т.д.

Один из методов решения данной задачи основан на так называемой теореме Форда и Фалкерсона, которая называется ещё теоремой о максимальном потоке и минимальном разрезе.

Теорема. Для любой сети максимальная величина потока из s в t равна минимальной пропускной способности разреза, отделяющего s от t .

Дадим *определение разреза*. Рассмотрим два множества вершин сети: $X \subset V$ и $\bar{X} \subset V$, где $\bar{X} = V \setminus X$, причём $s \in X$, а $t \in \bar{X}$. *Разрез N-N* в сети, отделяющий узлы s и t , есть множество дуг (x, \bar{x}) , где $x \in X$, а $\bar{x} \in \bar{X}$. Просуммировав все пропускные способности этих дуг $c(x, \bar{x})$, получим *пропускную способность* данного *разреза*. Очевидно, что любая цепь из s в t содержит, по крайней мере, одну дугу каждого разреза.

Описание алгоритма Форда-Фалкерсона. Алгоритм состоит из двух этапов – насыщения потока и его перераспределения.

Насыщенной называется дуга, поток которой равен её пропускной способности. *Поток* называется *насыщенным*, если любая цепь из s в t содержит насыщенную дугу.

Этап 1. Разыскиваются всевозможные цепи из источника в сток. Все дуги цепи насыщаются одинаковым возможно большим потоком, который определяется пропускной способностью наиболее «тонкой» дуги или наименьшей разностью между пропускной способностью и потоком в дуге. Насыщенные дуги условно разрываются.

Различные цепи могут иметь общие дуги. Полученный поток будет согласован с *условием сохранения в узлах*: входящий в данную вершину поток равен потоку, выходящему из этой вершины. Поток в сети проходит по цепям, из источника в сток, то есть *исключен многократный проход* по отдельной дуге.

Первый этап считается завершённым, если не осталось ненасыщенных цепей из источника в сток. Получаем насыщенную и «разорванную» сеть. Решение, полученное на первом этапе, может оказаться не единственным.

Этап 2. Начинаем перераспределение потока исходя из условия достижения общего по сети максимума потока. Для этого на части графа, состоящей из «неразорванных» дуг, мысленно снимается ориентация дуг и разыскиваются цепи из источника в сток, которые состоят из ненасыщенных дуг, имеющих направление от s к t (вперед), и непустых дуг, направленных назад.

Потоки в дугах прямого направления увеличиваются на такое же количество единиц, на которое уменьшаются потоки в дугах обратного направления выбранной цепи. Очевидно, что при этом нельзя превышать пропускную способность дуг прямого направления и допускать отрицательные потоки в дугах обратного направления.

Иногда при удачном выборе цепей на первом этапе решения можно обойтись без перераспределения потока, то есть максимальный поток на сети получается на первом этапе.

Задача 18.3.1. Задана пропускная способность дуг сети с началом в узле 1 и концом в узле 8 (рис. 18.10). Найти максимальный поток по сети, используя алгоритм Форда-Фалкерсона.

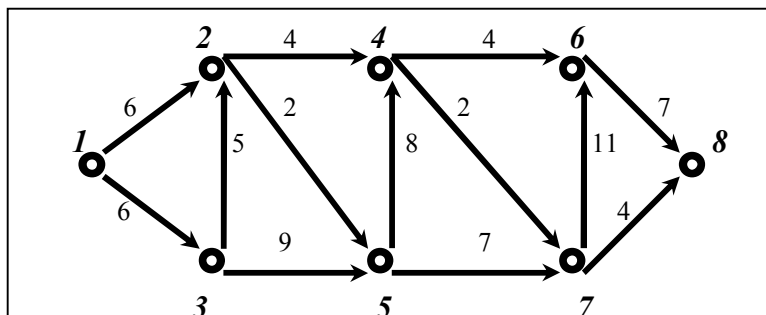


Рис. 18.10

Решение.

Этап 1 – насыщение потока.

Рассмотрим цепь 1-2-4-6-8. Наибольший поток, который можно пропустить через неё, равен 4. При этом дуги (2,4) и (4,6) станут насыщенными. Условно разрываем их. Получим новую сеть (рис. 18.11).

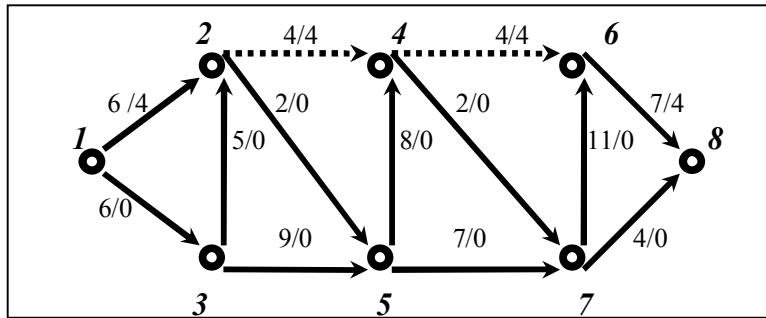


Рис. 18.11

Рассмотрим цепь $1-3-5-7-8$. Наибольший поток, который можно пропустить через неё, также равен 4. При этом дуга $(7,8)$ насытится. Распределение потока покажем на графе (рис. 18.12).

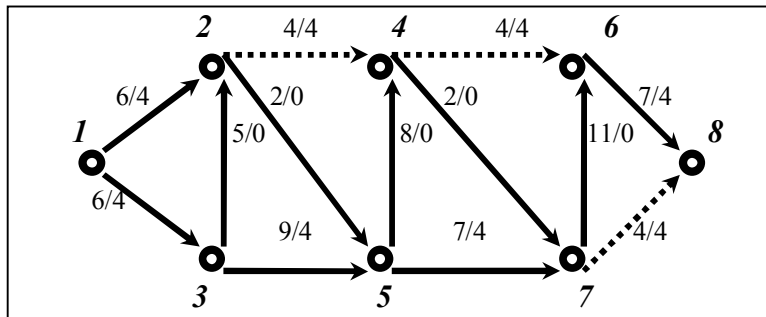


Рис. 18.12

Рассмотрим цепь $1-3-2-5-4-7-6-8$. Можно увеличить поток на 2. При этом насытятся дуги $(1,3)$, $(2,5)$, $(4,7)$. Покажем новое распределение потока на рис. 18.13.

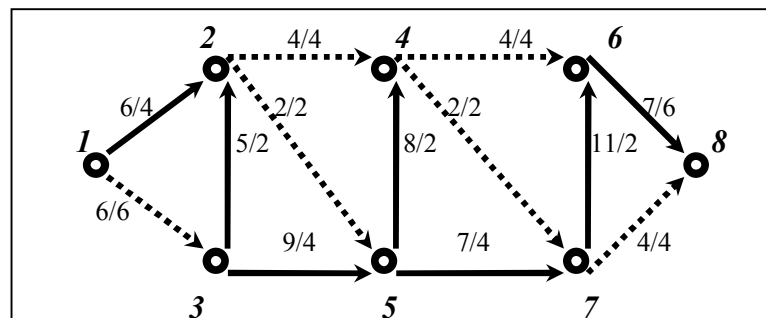


Рис. 18.13

Из 1 в 8 больше нет ненасыщенных цепей. По дуге $(1,3)$ двигаться нельзя, так как она уже насыщена. Движение по $(1,2)$ заканчивается в вершине 2 , так как обе выходящие из неё дуги насыщены. Переходим к этапу 2.

Этап 2 – перераспределение потока.

На сети (рис. 18.13) мысленно снимем ориентацию с ненасыщенных дуг и находим последовательность вершин от 1 к 8 (из источника в сток) такую, что дуги, соединяющие соседние вершины прямого направления, не насыщены, а дуги обратного направления не пусты.

Имеем единственную последовательность таких вершин: $1-2-3-5-7-6-8$ (рис. 18.13). «Вытянем» эту цепь в линию, сохраняя ориентацию дуг (рис. 18.14).

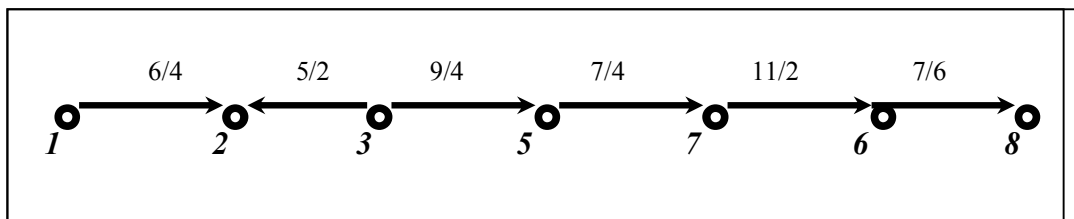


Рис. 18.14

Определим на дугах новый поток, увеличивая его на 1 на дугах, ориентированных в направлении движения «от источника к стоку», и, соответственно, уменьшая поток на 1 на дугах, направленных против этого движения. Процесс продолжаем до тех пор, пока какая-либо из прямых дуг цепи не станет насыщенной или какая-то из обратных дуг не станет пустой (рис. 18.15).

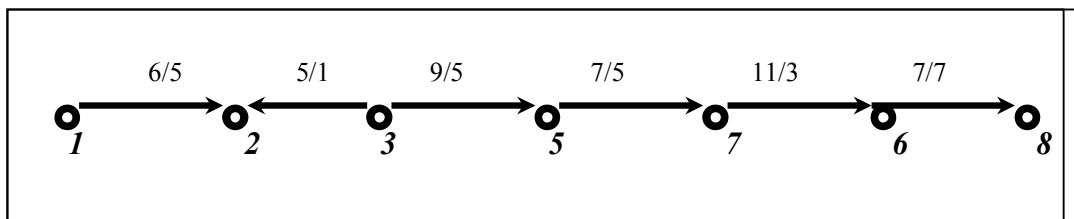


Рис. 18.15

Дуга (6,8) стала насыщенной. Дальнейшее увеличение потока по данному направлению невозможно.

Внесём новые данные в схему сети (рис. 18.16).

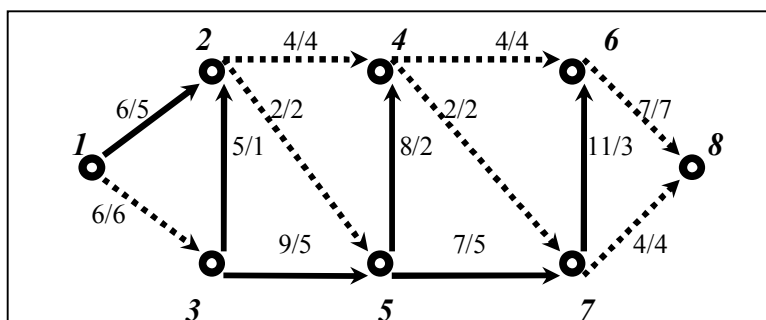


Рис. 18.16

Совершенно очевидно, что дальнейшее перераспределение потока невозможно, так как все дуги, ведущие в сток **8** из смежных вершин **6** и **7**, являются насыщенными. Достигнут максимум потока на сети, величину которого можно вычислить по исходящему из источника потоку или по входящему в сток. Очевидно, что эти числа должны быть равны:

$$f = \varphi(1,2) + \varphi(1,3) = 5 + 6 = 11; \quad f = \varphi(6,8) + \varphi(7,8) = 7 + 4 = 11.$$

Для проверки решения нужно проверить условие сохранения потока по узлам: для каждого узла суммарный входящий должен быть равен суммарному выходящему потоку. Например, для узла **2** входящий равен $5+1=6$, выходящий $2+4=6$. Для узла **3**: входящий 6 , выходящий $1+5=6$ и т.д. (рис. 18.17).

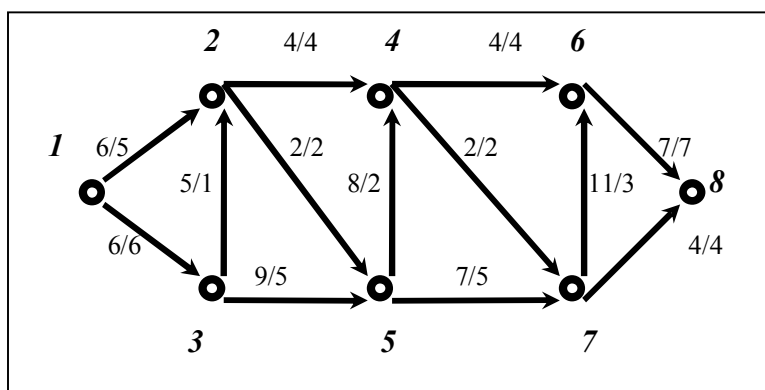


Рис. 18.17

Как отмечалось ранее, распределение потока по дугам при одном и том же суммарном максимальном потоке не единственное.

Задачи для самостоятельного решения.

1. Для рассмотренной задачи найти другое распределение максимального потока по дугам.
2. Найти максимальный поток в сети (рис. 18.18 и рис. 18.19).

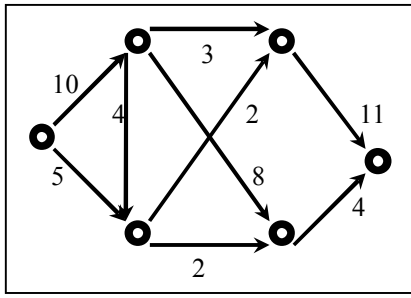


Рис. 18.18

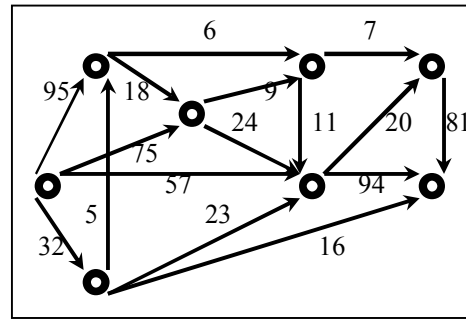


Рис. 18.19

ЛИТЕРАТУРА

1. Міхайленко В.М., Федоренко Н.Д., Демченко В.В. Дискретна математика. – К.:Видавництво Європейського університету, 2003. – 317 с.
2. Мельников В.Н. Логические задачи. – Киев – Одесса.: Вища школа, 1989. – 344 с.
3. Остапович М.В., Чернишенко С.В., Ротар О.А. Дискретна математика для інформатиків. – Д.: видавництво ДНУ, 2008. – 183 с.
4. Тишин В.В. Дискретная математика в примерах и задачах. – Спб.: БХВ–Петербург, 2008. – 354 с.
5. Хаггарти Р. Дискретная математика для программистов. – М.: Техносфера, 2005. – 400 с.
6. Асеев Г.Г., Абрамов О.М., Ситников Д.Э. Дискретная математика. – К.: Кондор, 2008. – 162 с.
7. Балюкевич Э.Л., Ковалёва Л.Ф., Романников А.Н. Дискретная математика. – М.: Московский государственный университет экономики, статистики и информатики, 2007. – 125 с.
8. Шевелёв Ю.П. Дискретная математика. – Т.: Томский государственный университет систем управления и радиоэлектроники, 2003. – 118 с.
9. Кулабухов С.Ю. Дискретная математика. – Шахты: Южно-российский государственный университет сервиса, 2006. – 150 с.
10. Оре О. Теория графов. – М.: Наука, 1980. – 352 с.
11. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
12. Форд Л., Фалкерсон Д. Потоки в сетях. – М.: Мир, 1966. – 276 с.

13. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергия. 1980. – 341 с.
14. Кузнецов О.П. Основы дискретной математики. Видеокурс лекций. Интернет-университет информационных технологий: www.INTUIT.ru
15. Флеров Ю.А. Дискретный анализ. Видеокурс лекций. Интернет-университет информационных технологий: www.INTUIT.ru

Навчальне видання

Швачич Геннадій Григорович

Бартенєв Георгій Михайлович

Онищенко Олександр Васильович

Толстой Віктор Володимирович

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

Частина III. Основи теорії графів.

Навчальний посібник

(російською мовою)

Тем. план 2014, поз. 271

Підписано до друку 22.04.2014. Формат 60x84 1/16. Папір друк. Друк плоский.
Облік.-вид. арк. 4,24. Умов. друк. арк. 4,18. Тираж 100 пр. Замовлення №

Національна металургійна академія України
49600, м. Дніпропетровськ-5, пр. Гагаріна,4

Редакційно-видавничий відділ НМетАУ