Министерство образования РФ
Рязанская государственная радиотехническая академия

Дискретная математика

Методические указания к лабораторным работам

Рязань 2004

УДК 519.17

Дискретная математика: Методические указания к лабораторным работам / Рязанская государственная радиотехническая академия; Сост. А.М. Гостин. Рязань, 2004. 24 с.

Содержит описания четырех лабораторных работ, посвященных решению прикладных задач теории графов.

Предназначены для студентов специальности 220300.

Табл. Ил. Библиогр.: назв.

Печатается по решению методического совета Рязанской государственной радиотехнической академии.

Рецензент: кафедра САПР ВС Рязанской государственной радиотехнической академии (зав. кафедрой проф. Корячко В.П.)

Дискретная математика

Составитель: Гостин Алексей Михайлович

Редактор
Корректор

Подписано в печать Формат бумаги 60 × 84 1/16.
Бумага газетная. Печать ротапринтная. Усл. печ. Л. 1,5.
Уч.-изд. Л. 1,5. Тираж 50 экз. Заказ

Рязанская государственная радиотехническая академия
390005, Рязань, ул. Гагарина 59/1.

ЛАБОРАТОРНАЯ РАБОТА №1

МАТРИЧНЫЕ СПОСОБЫ ПРЕДСТАВЛЕНИЯ ГРАФОВ

1 ЦЕЛЬ РАБОТЫ

Целью работы является изучение матричных способов представления графов.

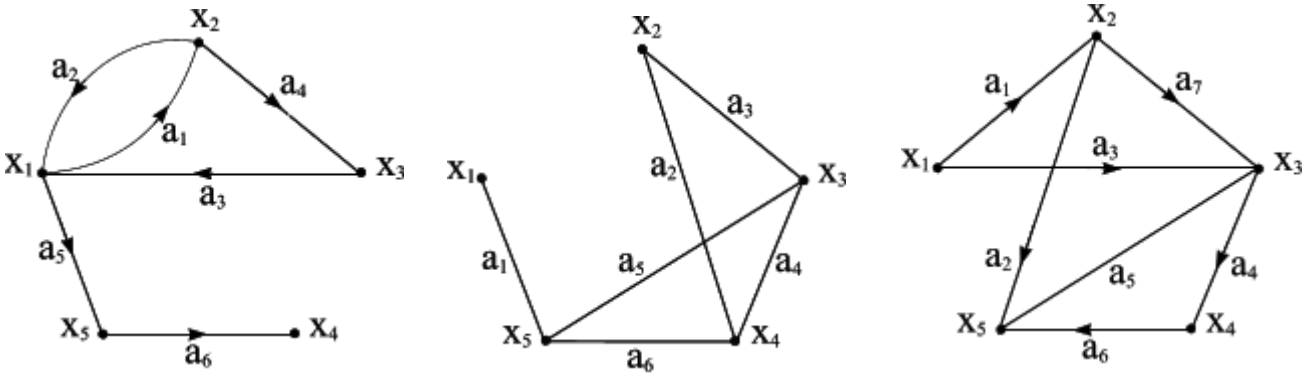
2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В последнее время теория графов стала простым, доступным и мощным средством решения вопросов, относящихся к широкому кругу проблем. Это проблемы проектирования интегральных схем и схем управления, исследования автоматов, логических цепей, блок-схем программ, экономики и статистики, химии и биологии, теории расписаний и дискретной оптимизации.

2.1 ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Граф G задается множеством точек или вершин $x_1,x_2,\ldots,x_n$ (которое обозначается через X) и множеством линий или *ребер* $a_1,a_2,\ldots,a_n$ (которое обозначается символом A), соединяющих между собой все или часть этих точек. Таким образом, граф G полностью задается (и обозначается) парой (X, A).

Если ребра из множества A ориентированы, что обычно показывается стрелкой, то они называются *дугами*, и граф с такими ребрами называется *ориентированным* графом (рисунок 1(а)). Если ребра не имеют ориентации, то граф называется *неориентированным* (рисунок 1(б)). В случае когда $G=(X, A)$ является ориентированным графом и мы хотим пренебречь направленностью дуг из множества A, то неориентированный граф, соответствующий G, будем обозначать как $G=(X, A)$.



(а) (б) (в)

Рисунок 1 (а) – ориентированный граф; (б) – неориентированный граф; (в) – смешанный граф.

Если дуга обозначается упорядоченной парой, состоящей из *начальной* и *конечной* вершин (т. е. двумя *концевыми* вершинами дуги), ее направление

предполагается заданным от первой вершины ко второй. Так, например, на рисунке 1(а) обозначение (x_1,x_2) относится к дуге a_1, а (x_2,x_1) - к дуге a_2.

Другое, употребляемое чаще описание ориентированного графа G состоит в задании множества вершин X и *соответствия* Γ, которое показывает, как между собой связаны вершины. Соответствие Γ называется *отображением* множества X в X, а граф в этом случае обозначается парой $G=(X,\Gamma)$.

Для графа на рисунке 1(а) имеем $\Gamma(x_1)=\{x_2,x_5\}$, т. е. вершины x_2 и x_5 являются конечными вершинами дуг, у которых начальной вершиной является x_1.

$\Gamma(x_2)=\{x_1,x_3\}$, $\Gamma(x_3)=\{x_1\}$, $\Gamma(x_4)=\varnothing$ - пустое множество, $\Gamma(x_5)=\{x_4\}$.

В случае неориентированного графа или графа, содержащего и дуги, и неориентированные ребра (см., например, графы, изображенные на рисунках 1(б) и 1(в)), предполагается, что соответствие Γ задает такой эквивалентный ориентированный граф, который получается из исходного графа заменой каждого неориентированного ребра двумя противоположно направленными дугами, соединяющими те же самые вершины. Так, например, для графа, приведенного на рисунке 1(б), имеем $\Gamma(x_5)=\{x_1,x_3,x_4\}$, $\Gamma(x_1)=\{x_5\}$ и др.

Поскольку *прямое соответствие* или *образ* вершины $\Gamma(x_i)$ представляет собой множество таких вершин $x_j\in X$, для которых в графе G существует дуга (x_i,x_j), то через $\Gamma^{-1}(x_i)$ естественно обозначить множество вершин x_k, для которых в G существует дуга (x_k,x_i). Такое отношение принято называть *обратным соответствием* или *прообразом* вершины. Для графа, изображенного на рисунке 1(а), имеем

$\Gamma^{-1}(x_1)=\{x_2,x_3\}$, $\Gamma^{-1}(x_2)=\{x_1\}$ и т. д.

Вполне очевидно, что для неориентированного графа $\Gamma^{-1}(x_i)=\Gamma(x_i)$ для всех $x_i\in X$.

Когда отображение Γ действует не на одну вершину, а на множество вершин $X_q=\{x_1,x_2,\ldots,x_q\}$, то под $\Gamma(X_q)$ понимают объединение $\Gamma(x_1)\cup\Gamma(x_2)\cup\ldots\cup\Gamma(x_q)$, т. е. $\Gamma(X_q)$ является множеством таких вершин $x_j\in X$, что для каждой из них существует дуга (x_i,x_j) в G, где $x_i\in X_q$. Для графа, приведенного на рисунке 1(а), $\Gamma(\{x_2,x_5\})=\{x_1,x_3,x_4\}$ и $\Gamma(\{x_1,x_3\})=\{x_2,x_5,x_1\}$.

Отображение $\Gamma(\Gamma(x_i))$ записывается как $\Gamma^2(x_i)$. Аналогично "тройное" отображение $\Gamma(\Gamma(\Gamma(x_i)))$ записывается как $\Gamma^3(x_i)$ и т. д. Для графа, показанного на рисунке 1(а), имеем:

$\Gamma^2(x_1)=\Gamma(\Gamma(x_1))=\Gamma(\{x_2,x_5\})=\{x_1,x_3,x_4\}$;

$\Gamma^3(x_1)=\Gamma(\Gamma^2(x_1))=\Gamma(\{x_1,x_3,x_4\})=\{x_2,x_5,x_1\}$ и т. д.

Аналогично понимаются обозначения $\Gamma^{-2}(x_i)$, $\Gamma^{-3}(x_i)$ и т. д.

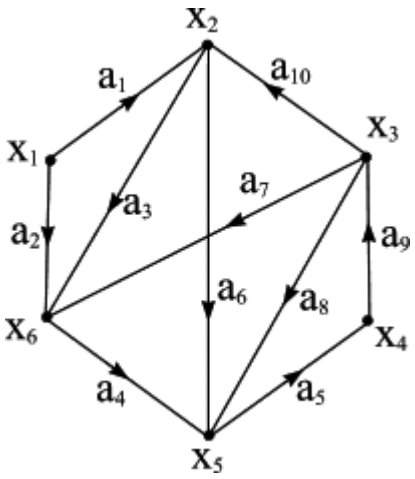


Рисунок 2.

Дуги $a=(x_i,x_j)$, $x_i \neq x_j$, имеющие общие концевые вершины, называются *смежными*. Две вершины x_i и x_j называются смежными, если какая-нибудь из двух дуг (x_i,x_j) и (x_j,x_i) или обе одновременно присутствуют в графе. Так, например, на рисунке 2 дуги a_1, a_{10}, a_3 и a_6 как и вершины x_5 и x_3, являются смежными, в то время как дуги a_1 и a_5 или вершины x_1 и x_4 не являются смежными.

Число дуг, которые имеют вершину x_i своей начальной вершиной, называется *полустепенью исхода* вершины x_i, и, аналогично, число дуг, которые имеют x_i своей конечной вершиной, называется *полустепенью захода* вершины x_i.

Таким образом, на рисунке 2 полустепень исхода вершины x_3, обозначаемая через $deg^+(x_3)$, равна $|\Gamma(x_3)|=3$, и полустепень захода вершины x_3, обозначаемая через $deg^-(x_3)$, равна $|\Gamma^{-1}(x_3)|=1$.

Очевидно, что сумма полустепеней захода всех вершин графа, а также сумма полустепеней исхода всех вершин равны общему числу дуг графа G, т. е.

$$\sum_{i=1}^{n} \deg^+(x_i) = \sum_{i=1}^{n} \deg^-(x_i) = m, \qquad (1)$$

где n - число вершин и m - число дуг графа G.

Для неориентированного графа $G=(X,\Gamma)$ *степень* вершины x_i определяется аналогично - с помощью соотношения $deg(x_i) \equiv |\Gamma(x_i)| = |\Gamma^{-1}(x_i)|$.

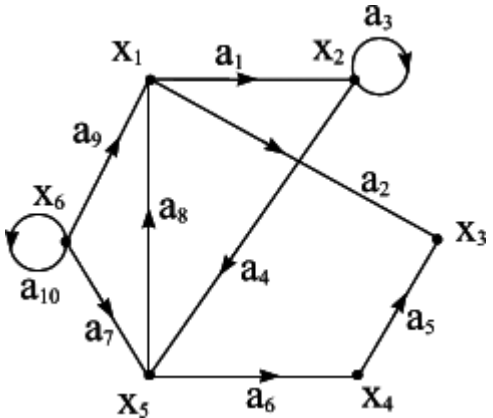
Петлей называется дуга, начальная и конечная вершины которой совпадают. На рисунке 3, например, дуги a_3 и a_{10} являются петлями.

2.2 МАТРИЧНЫЕ ПРЕДСТАВЛЕНИЯ

2.2.1 МАТРИЦА СМЕЖНОСТИ

Пусть дан граф G, его матрица смежности обозначается через $\mathbf{A}=[a_{ij}]$ и определяется следующим образом:

$a_{ij}=1$, если в G существует дуга (x_i,x_j),
$a_{ij}=0$, если в G нет дуги (x_i,x_j).



$$A = \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ x_2 & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ x_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ x_4 & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}$$

Таким образом,

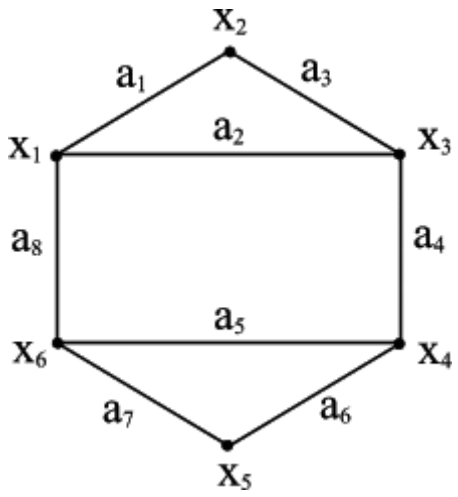
матрица смежности графа, изображенного на рисунке 3, имеет вид

Рисунок 3.

Матрица смежности полностью определяет структуру графа. Например, сумма всех элементов строки x_i матрицы дает полустепень исхода вершины x_i, а сумма элементов столбца x_i - полустепень захода вершины x_i. Множество столбцов, имеющих 1 в строке x_i есть множество $\Gamma(x_i)$, а множество строк, которые имеют 1 в столбце x_i совпадает с множеством $\Gamma^{-1}(x_i)$.

Петли на графе представляют собой элементы, имеющие 1 на главной диагонали матрицы, например a_{22}, a_{66} для графа, изображенного на рисунке 3.

В случае неориентированного графа матрица смежности является симметричной относительно главной диагонали (рисунок 4).



A=

	x1	x2	x3	x4	x5	x6
x1	**0**	**1**	**1**	**0**	**0**	**1**
x2	**1**	**0**	**1**	**0**	**0**	**0**
x3	**1**	**1**	**0**	**1**	**0**	**0**
x4	**0**	**0**	**1**	**0**	**1**	**1**

Рисунок 4.

2.2.2 МАТРИЦА ИНЦИДЕНТНОСТИ

Пусть дан граф G с n вершинами и m дугами. *Матрица инцидентности* графа G обозначается через $\mathbf{B}=[b_{ij}]$ и является матрицей размерности $n \times m$, определяемой следующим образом:

$b_{ij}=1$, если x_i является начальной вершиной дуги a_j;
$b_{ij}=-1$, если x_i является конечной вершиной дуги a_j;
$b_{ij}=0$, если x_i не является концевой вершиной дуги a_j.

Для графа, приведенного на рисунке 3, матрица инцидентности имеет вид:

		a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
	x_1	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**-1**	**-1**	**0**
	x_2	**-1**	**0**	**±1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
B=	x_3	**0**	**-1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	x_4	**0**	**0**	**0**	**0**	**-1**	**-1**	**0**	**0**	**0**	**0**

Поскольку каждая дуга инцидентна двум различным вершинам (за исключением случая, когда дуга образует петлю), то каждый столбец содержит один элемент, равный 1, и один - равный -1. Петля в матрице инцидентности не имеет адекватного математического представления (в программной реализации допустимо задание одного элемента b_{ij}=1).

Если G является неориентированным графом (рисунок 4), то его матрица инцидентности определяется следующим образом:

b_{ij}=1, если x_i является концевой вершиной дуги a_j;
b_{ij}=0, если x_i не является концевой вершиной дуги a_j.

		a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
	x_1	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**1**
	x_2	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
B=	x_3	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
	x_4	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**

Матрица инцидентности, как способ задания графов, успешно применяется при описании мультиграфов (графов, в которых смежные вершины могут соединяться несколькими параллельными дугами).

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Получить задание у преподавателя в виде одного из двух способов матричного представления графа:

а) матрица смежности; б) матрица инцидентности

3.2 Составить алгоритм программы, реализующей перевод из заданного способа матричного представления графа в другой, учитывая при этом исходный тип графа (неориентированный, ориентированный, смешанный).

3.3 Создать программу, реализующую перевод из заданного способа матричного представления графа в другой. Предусмотреть консольный ввод исходных данных и вывод результатов работы программы на экран.

4 СОДЕРЖАНИЕ ОТЧЕТА ПО РАБОТЕ

4.1 Исходное задание и цель работы.
4.2 Блок-схема программы по п.3.2.
4.3 Распечатка текста программы по п.3.3.
4.4 Контрольный пример и результаты машинного расчета.
4.5 Выводы по работе.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Перечислите основные способы представления графов.
5.2 Покажите на примере прямое и обратное соответствие для заданной вершины.
5.3 Чему равна сумма степеней всех вершин неориентированного графа?
5.4 В чем отличия матричного представления ориентированных и неориентированных графов?
5.5 В чем особенности представления графа матрицей смежности?
5.6 В чем особенности представления графа матрицей инцидентности?

ЛАБОРАТОРНАЯ РАБОТА №2

ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ

1 ЦЕЛЬ РАБОТЫ

Целью работы является изучение алгоритмов поиска кратчайших путей на графах на примере метода динамического программирования.

2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Путем (или *ориентированным маршрутом*) ориентированного графа называется последовательность дуг, в которой конечная вершина всякой дуги, отличной от последней, является начальной вершиной следующей. Так на рисунке 5 последовательности дуг $\mu_1=\{a_6, a_5, a_9, a_8, a_4\}$, $\mu_2=\{a_1, a_6, a_5, a_9\}$, $\mu_3=\{a_1, a_6, a_5, a_9, a_{10}, a_6, a_4\}$ являются путями.

Ориентированной цепью (*орцепью*) называется такой путь, в котором каждая дуга используется не больше одного раза. Так, например, приведенные выше пути μ_1 и μ_2 являются орцепями, а путь μ_3 не является таким, поскольку дуга a_6 в нем используется дважды.

Маршрут есть неориентированный "двойник" пути, и это понятие рассматривается в тех случаях, когда можно пренебречь направленностью дуг в графе.

Таким образом, маршрут есть последовательность ребер a_1, a_2,...,a_q, в которой каждое ребро a_i, за исключением, возможно, первого и последнего ребер, связано с ребрами a_{i-1} и a_{i+1} своими двумя концевыми вершинами.

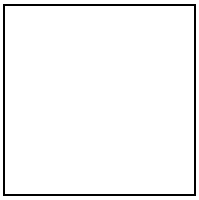


Рисунок 5.

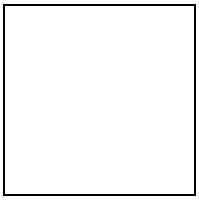


Рисунок 6.

Последовательности дуг на рисунке 6

$\mu_4=\{a_2, a_4, a_8, a_{10}\}$, $\mu_5=\{a_2, a_7, a_8, a_4, a_3\}$ и $\mu_6=\{a_{10}, a_7, a_4, a_8, a_7, a_2\}$

являются маршрутами.

Контуром (простой цепью) называется такой путь (маршрут), в котором каждая вершина используется не более одного раза. Например, путь μ_2 является контуром, а пути μ_1 и μ_3 - нет. Очевидно, что контур является также цепью, но обратное утверждение неверно. Например, путь μ_1 является цепью, но не контуром, путь μ_2 является цепью и контуром, а путь μ_3 не является ни цепью, ни контуром.

Аналогично определяется простая цепь в неориентированных графах. Так, например, маршрут μ_4 есть простая цепь, маршрут μ_5 - цепь, а маршрут μ_6 не является цепью.

Путь или маршрут можно изображать также последовательностью вершин. Например, путь μ_1 можно представить также: $\mu_1=\{x_2, x_5, x_4, x_3, x_5, x_6\}$ и такое представление часто оказывается более полезным в тех случаях, когда осуществляется поиск контуров или простых цепей.

Иногда дугам графа G сопоставляются (приписываются) числа - дуге (x_i,x_j) ставится в соответствие некоторое число c_{ij}, называемое *весом*, или *длиной*, или *стоимостью* (*ценой*) дуги. Тогда граф G называется *взвешенным*. Иногда веса (числа v_i) приписываются вершинам x_i графа.

При рассмотрении пути μ, представленного последовательностью дуг (a_1, a_2,...,a_q), за его *вес* (или *длину*, или *стоимость*) принимается число $L(\mu)$, равное сумме весов всех дуг, входящих в μ, т. е.

$$L(\mu) = \sum_{(x_i, x_j) \in \mu} c_{ij} \qquad (2)$$

Таким образом, когда слова "длина", "стоимость", "цена" и "вес" применяются к дугам, то они эквивалентны по содержанию, и в каждом конкретном случае выбирается такое слово, которое ближе подходит по смыслу.

Длиной (или *мощностью*) пути и называется число дуг, входящих в него.

2.2 ЗАДАЧА О КРАТЧАЙШЕМ ПУТИ

Пусть дан граф $G=(X,\Gamma)$, дугам которого приписаны веса (стоимости), задаваемые матрицей $\mathbf{C}=[c_{ij}]$. *Задача о кратчайшем пути* состоит в нахождении кратчайшего пути от заданной начальной вершины (*истока*) s до заданной конечной вершины (*стока*) t, при условии, что такой путь существует:

Найти $\mu(s,t)$ при $L(\mu) \to min$, $s,t \in X$, $t \in R(s)$,

где $R(s)$ - множество, достижимое из вершины s.

В общем случае элементы c_{ij} матрицы весов $\mathbf{C}$ могут быть положительными, отрицательными или нулями. Единственное ограничение состоит в том, чтобы в G не было циклов с суммарным отрицательным весом. Отсюда следует, что дуги (ребра) графа G не должны иметь отрицательные веса.

Почти все методы, позволяющие решить задачу о кратчайшем (s-t)-пути, дают также (в процессе решения) и все кратчайшие пути от s к $x_i (\forall x_i \in X)$. Таким образом, они позволяют решить задачу с небольшими дополнительными вычислительными затратами.

Допускается, что матрица весов $\mathbf{C}$ не удовлетворяет условию треугольника, т. е. не обязательно $c_{ij} \le c_{ik} + c_{kj}$ для всех i, j и k.

Если в графе G дуга (x_i, x_j) отсутствует, то ее вес полагается равным ∞.

Ряд задач, например, задачи нахождения в графах путей с максимальной надежностью и с максимальной пропускной способностью, связаны с задачей о кратчайшем пути, хотя в них характеристика пути (скажем, вес) является не суммой, а некоторой другой функцией характеристик (весов) дуг, образующих путь. Такие задачи можно переформулировать как задачи о кратчайшем пути и решать их соответствующим образом.

Существует множество методов решения данной задачи, отличающиеся областью применимости и трудоемкостью (Дейкстры, Флойда, динамического программирования). Среди них большое распространение получили частные алгоритмы, применяющиеся при решении частных задач, и имеющие меньшую трудоемкость. Эти частные случаи встречаются на практике довольно часто

(например, когда c_{ij} являются расстояниями), так что рассмотрение этих специальных алгоритмов оправдано.

На практике задачу кратчайшего пути часто требуется решать для класса ориентированных ациклических графов. Такая задача успешно решается с помощью метода динамического программирования.

2.3 МЕТОД ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Прямая итерация. Пусть вершины пронумерованы так, что дуга (x_i,x_j) всегда ориентирована от вершины x_i к вершине x_j, имеющей больший номер. Для ациклического графа такая нумерация всегда возможна и производится очень легко. При этом начальная вершина s получает номер 1, а конечная t - номер n.

Пусть $\lambda(x_i)$ – пометка вершины x_i, равная длине кратчайшего пути от 1 до x_i, s – начальная вершина (источник), t – конечная вершина (сток).

Шаг 1. Положить $\lambda(s)=0$, $\lambda(x_i) = \infty$ для всех вершин $x_i \in X/s$; $i=1$;

Шаг 2. $i=i+1$. Присвоим вершине x_j пометку $\lambda(x_j)$, – равную длине кратчайшего пути от 1 до x_j, используя для этого соотношение

$$\lambda(x_j) = \min_{x_i \in \Gamma^{-1}(x_j)} [\lambda(x_i) + c_{ij}] \qquad (3)$$

Шаг 3. Повторить п.2. до тех пор, пока последняя вершина n не получит пометку $\lambda(t)$.

Необходимо отметить, что если вершина x_j помечена, то пометки $\lambda(x_i)$ известны для всех вершин $x_i \in \Gamma^{-1}(x_j)$, так как в соответствии со способом нумерации это означает, что $x_i<x_j$ и, следовательно, вершины x_i уже помечены в процессе применения алгоритма.

Пометка $\lambda(t)$ равна длине самого короткого пути от s до t. Сами дуги, образующие путь, могут быть найдены способом последовательного возвращения. А именно дуга (x_i,x_j), согласно (3), принадлежит пути тогда и только тогда, когда

$$\lambda(x_j)= \lambda(x_i)+c_{ij} \qquad (4)$$

Обратная итерация: начиная с вершины t, имеющей номер n, полагаем на каждом шаге x_j равной такой вершине (скажем, x_j^*), для которой выполняется соотношение (4), и так продолжаем до тех пор, пока не будет достигнута начальная вершина (т.е. пока не будет $x_j^* \equiv s$).

Совершенно очевидно, что пометка $\lambda(x_j)$ вершины x_j дает длину кратчайшего пути μ от s до x_j.

2.4 АЛГОРИТМ ТОПОЛОГИЧЕСКОЙ СОРТИРОВКИ

В некоторых случаях исходный граф является ациклическим, но имеет неправильную нумерацию – содержит дуги (x_j,x_i), ориентированные от вершины x_j к вершине x_i, имеющей меньший номер ($j>i$). Для успешного нахождения кратчайшего пути с помощью метода динамического

программирования к такому графу сначала применяется алгоритм топологической сортировки вершин.

Алгоритм топологической сортировки вершин очень простой. Он позволяет не только правильно перенумеровать вершины графа, но и определить его ацикличность.

Шаг 1. Положить $i=n$, где n – число вершин графа G.

Шаг 2. В графе определяется вершина x_k, для которой выполняется условие $|\Gamma(x_k)|=\varnothing$ (т.е., вершина, из которой не выходит ни одна дуга). Вершина x_k получает порядковый номер i (перенумеруется) и исключается из дальнейшего рассмотрения вместе со всеми входящими в нее инцидентными дугами. $i=i-1$.

Шаг 3. Повторять п.2. до тех пор, пока не будет выполнено одно из условий:

1) $i=1$ – достигнута начальная вершина. Вершины графа получили правильную нумерацию.
2) Невозможно определить вершину, для которой выполнялось бы условие $|\Gamma(x_k)|=\varnothing$. В графе имеется цикл.

В последнем случае алгоритм динамического программирования неприменим. Для поиска кратчайших путей на таком графе необходимо использовать более эффективные методы, например, алгоритм Дейкстры [].

2.5 КОНТРОЛЬНЫЙ ПРИМЕР

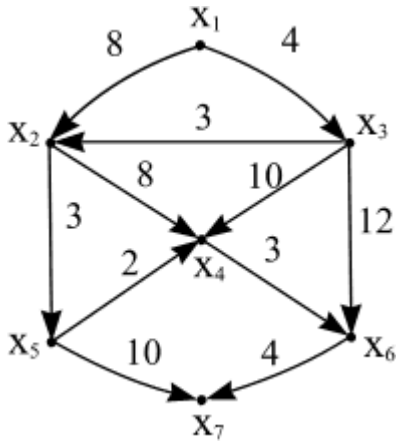


Рисунок 7.

Для графа, изображенного на рисунке 7, определим кратчайший путь между вершинами x_1 и x_7, используя метод динамического программирования.

Так как граф содержит дуги (x_3 ,x_2) и (x_5 ,x_4), имеющие неправильную нумерацию (от большего к меньшему), необходимо перенумеровать вершины графа, применяя алгоритм топологической сортировки вершин.

В нашем случае из графа будут последовательно исключаться вершины x_7, x_6 , x_4, x_5, x_2, x_3, x_1. Соответственно, вершины графа получат новую нумерацию, и граф будет иметь вид, представленный на рисунке 8 (старая нумерация вершин сохранена в скобках).

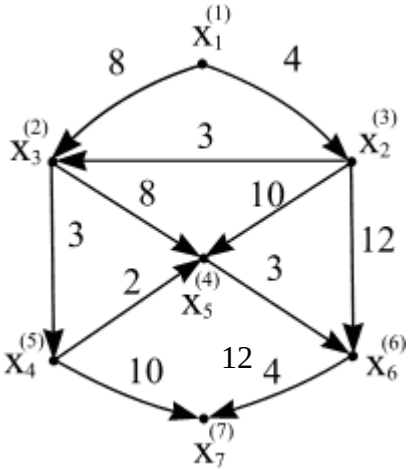


Рисунок 8.

На первом шаге оценка $\lambda(x_1)=0$. Переходим к вершине x_2. Множество $\Gamma^{-1}(x_2)$

включает только одну вершину x_1. Следовательно, оценка для вершины x_2 определяемая по формуле (3), будет $\lambda(x_2)=min\{0+4\}=4$. Переходим к вершине x_3. Для вершины x_3 множество $\Gamma^{-1}(x_3)=\{x_1, x_2\}$. В этом случае, оценка будет выбираться как минимальная из двух возможных: $\lambda(x_3)=min\{0+8, 4+3\}=7.$ Для вершины x_4 оценка определяется снова однозначно: $\lambda(x_4)=min\{7+3\}=10$. Однако при переходе к вершине x_5 мы получаем сразу три входящих дуги (x_2, x_5), (x_3, x_5), (x_4, x_5). Применяя формулу (3), определяем оценку для вершины x_5:

$$\lambda(x_5)=min\{4+10, 7+8, 10+2\}=12.$$

Далее, аналогичным образом вершина x_6 получает оценку $\lambda(x_6)=min\{4+12, 12+3\}=15$ и, наконец, вершина x_7 получает оценку $\lambda(x_7)=min\{10+10, 15+4\}=19$.

Таким образом, конечная вершина x_7 пути $\mu(x_1,x_7)$ достигнута, длина пути равна $L(\mu)=19.$

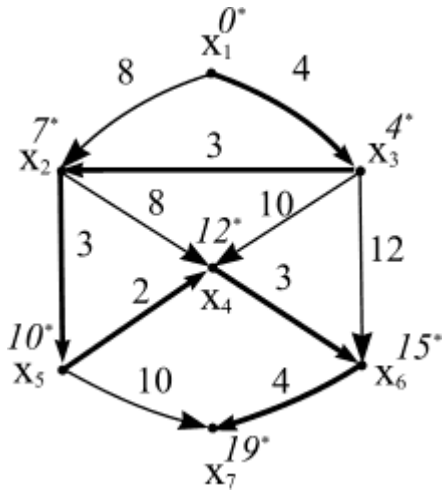


Рисунок 9.

Применяя выражение (4), последовательно определяем вершины, которые входят в кратчайший путь. Перемещаясь от конечной вершины x_7, выбираем последовательность вершин, для которой выражение (4) принимает значение «истинно»: $x_6, x_5, x_4, x_3, x_2, x_1$, т.е. кратчайший путь проходит последовательно через все вершины графа.

Действительно, легко убедиться в истинности выражений:

$\lambda(x_7)= \lambda(x_6)+ c_{67}$, $(19=15+4)$;
$\lambda(x_6)= \lambda(x_5)+ c_{56}$, $(15=12+3)$;
$\lambda(x_5)= \lambda(x_4)+ c_{45}$, $(12=10+2)$;
$\lambda(x_4)= \lambda(x_3)+ c_{34}$, $(10=7+3)$;
$\lambda(x_3)= \lambda(x_2)+ c_{23}$, $(7=4+3)$;
$\lambda(x_2)= \lambda(x_1)+ c_{12}$, $(4=0+4)$;

Произведя перенумерацию вершин графа на исходную, окончательно определяем кратчайший путь:

$\mu(x_1,x_7)=\{x_1, x_3, x_2, x_5, x_4, x_6, x_7\}$.

Задача решена.

Результат решения в виде выделенного пути изображен на рисунке 9. Курсивом «со звездочкой» отмечены значения пометок вершин $\lambda(x_i)$.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Получить задание у преподавателя в виде исходного ориентированного графа.

3.2 Составить блок-схему программы, определяющей кратчайший путь на графе от заданной начальной вершины s до заданной конечной вершины t с помощью метода динамического программирования.

3.3 Составить блок-схему программы, реализующей алгоритм топологической сортировки с произвольной нумерацией вершин графа.

3.4 Создать программу, реализующую метод динамического программирования и алгоритм топологической сортировки вершин. Исходный граф задается в виде матрицы смежности, вводимой построчно с помощью консоли. Указание: для определения вершин, входящих в множество $\Gamma^{-1}(x_i)$ используйте j-й столбец матрицы смежности.

4 СОДЕРЖАНИЕ ОТЧЕТА ПО РАБОТЕ

4.1 Исходное задание и цель работы.
4.2 Блок-схема программы по п.3.2.
4.3 Блок-схема программы по п.3.3.
4.3 Распечатка текста программы по п.3.4.
4.4 Контрольный пример и результаты машинного расчета.
4.5 Выводы по работе.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Дайте определение пути, маршрута, цепи, контура.
5.2 Какой граф называется взвешенным?
5.3 Как определяется длина пути графа?
5.4 Задача нахождения кратчайшего пути на графе.
5.5 Реализация метода динамического программирования для нахождения кратчайшего пути на графе.
5.6 Ограничения применения метода динамического программирования для нахождения кратчайшего пути на графе.
5.7 Что называется правильной нумерацией вершин графа?
5.8 Применение алгоритма топологической сортировки для перенумерации вершин графа.

ЛАБОРАТОРНАЯ РАБОТА №3

ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА

1 ЦЕЛЬ РАБОТЫ

Целью работы является изучение метода построения кратчайших остовых деревьев графа на примере алгоритма Прима-Краскала.

2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Одним из наиболее важных понятий теории графов является *дерево*.

Неориентированным деревом называется связанный граф, не имеющий циклов.

Суграфом графа G является подграф G_p, содержащий все вершины исходного графа.

Если $G=(X,A)$ - неориентированный граф с n вершинами, то связный суграф G_p, не имеющий циклов, называется *остовым деревом* (*остовом*) *графа* G.

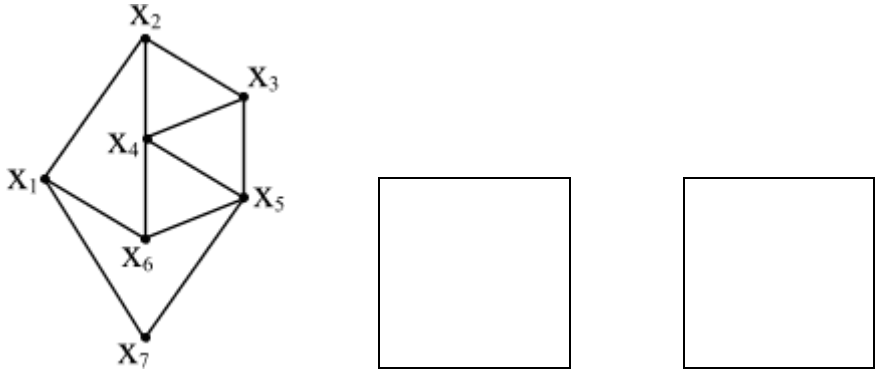
Для остового дерева справедливо соотношение:

$$G_p=(X_p,A_p) \subseteq G, \quad \text{где } X_p = X,\ A_p \subseteq A \tag{5}$$

Легко доказать, что остовое дерево имеет следующие свойства:

1) остовое дерево графа с n вершинами имеет $n-1$ ребро ($|X_p|=|A_p|-1$);
2) существует единственный путь, соединяющий любые две вершины остова графа:
   $\forall x_i,x_j \in X_p\ (i \neq j) \rightarrow \exists!\ \mu(x_i,x_j)$.

Например, если G - граф, показанный на рисунке 10(а), то графы на рисунках 10(б,в) являются остовами графа G. Из сформулированных выше определений вытекает, что остов графа G можно рассматривать как минимальный связанный остовый подграф графа G.



(а) (б) (в)

Рисунок 10 – Представление графа в виде остовых деревьев.

Понятие дерева как математического объекта было впервые предложено Кирхгофом в связи с определением фундаментальных циклов, применяемых при анализе электрических цепей. Приблизительно десятью годами позже Кэли вновь (независимо от Кирхгофа) ввел понятие дерева и получил большую часть первых результатов в области исследования свойств деревьев. Большую известность получила его знаменитая теорема:

Теорема Кэли. На графе с n вершинами можно построить n^{n-2} остовых деревьев.

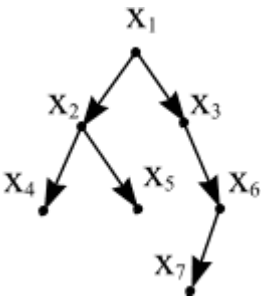


Рисунок 11.

Ориентированное дерево представляет собой ориентированный граф без циклов, в котором полустепень захода каждой вершины, за исключением одной (вершины r), равна единице, а полустепень захода вершины r (называемой корнем этого дерева) равна нулю.

На рисунке 11 показан граф, который является ориентированным деревом с корнем в вершине x_1. Из приведенного определения следует, что ориентированное дерево с n вершинами имеет $n-1$ дуг и связно.

Неориентированное дерево можно преобразовать в ориентированное: надо взять его произвольную вершину в качестве корня и ребрам приписать такую ориентацию, чтобы каждая вершина соединялась с корнем (только одной) простой цепью.

"Генеалогическое дерево", в котором вершины соответствуют лицам мужского пола, а дуги ориентированы от родителей к детям, представляет собой хорошо известный пример ориентированного дерева. Корень в этом дереве соответствует "основателю" рода (лицу, родившемуся раньше остальных).

2.2 КРАТЧАЙШИЙ ОСТОВ ГРАФА

В лабораторной работе исследуется метод прямого построения *кратчайших остовых деревьев* во взвешенном графе (в котором веса приписаны дугам). Кратчайшее остовое дерево графа находит применение при прокладке дорог (газопроводов, линий электропередач и т. д.), когда необходимо связать n точек некоторой сетью так, чтобы общая длина "линий связи" была минимальной. Если точки лежат на евклидовой плоскости, то их можно считать вершинами полного графа G с весами дуг, равными соответствующим "прямолинейным" расстояниям между концевыми точками дуг. Если "разветвление" дорог допускается только в заданных n точках, кратчайшее остовое дерево графа G будет как раз требуемой сетью дорог, имеющей наименьший вес.

Рассмотрим взвешенный связный неориентированный граф $G=(X,A)$; вес ребра (x_i,x_j) обозначим c_{ij}. Из большого числа остовов графа нужно найти один, у которого сумма весов ребер наименьшая. Такая задача возникает, например, в

том случае, когда вершины являются клеммами электрической сети, которые должны быть соединены друг с другом с помощью проводов наименьшей общей длины (для уменьшения уровня наводок). Другой пример: вершины представляют города, которые нужно связать сетью трубопроводов; тогда наименьшая общая длина труб, которая должна быть использована для строительства (при условии, что вне городов "разветвления" трубопроводов не допускаются), определяется кратчайшим остовом соответствующего графа.

Задача построения кратчайшего остова графа является одной из немногих задач теории графов, которые можно считать полностью решенными.

2.3 АЛГОРИТМ ПРИМА-КРАСКАЛА

Этот алгоритм порождает остовое дерево посредством разрастания только одного поддерева, например X_p, содержащего больше одной вершины. Поддерево постепенно разрастается за счет присоединения ребер (x_i,x_j), где $x_i \in X_p$ и $x_j \notin X_p$; причем добавляемое ребро должно иметь наименьший вес c_{ij}. Процесс продолжается до тех пор, пока число ребер в A_p не станет равным $n-1$. Тогда поддерево $G_p=(X_p,A_p)$ будет требуемым остовым деревом. Впервые такая операция была предложена Примом и Краскалом (с разницей – в способе построения дерева), поэтому данный алгоритм получил название Прима-Краскала.

Алгоритм начинает работу с включения в поддерево начальной вершины. Поскольку остовое дерево включает все вершины графа G, то выбор начальной вершины не имеет принципиального значения. Будем каждой очередной вершине присваивать пометку $\beta(x_i)=1$, если вершина x_i принадлежит поддереву X_p и $\beta(x_j)=0$ – в противном случае.

Алгоритм имеет вид:

Шаг 1. Пусть $X_p=\{x_1\}$, где x_1 - начальная вершина, и $A_p=\varnothing$ (A_p является множеством ребер, входящих в остовое дерево). Вершине x_1 присвоить пометку $\beta(x_1)=1$. Для каждой вершины $x_i \notin X_p$ присвоить $\beta(x_i)=0$.

Шаг 2. Из всех вершин $x_j \in \Gamma(X_p)$, для которых $\beta(x_j)=0$, найти вершину x_j^* такую, что

$$c(x_i,x_j^*)=\min_{x_j\in\Gamma(X_p)}\{c(x_i,x_j)\},\ \text{где } x_i\in X_p \text{ и } x_j\notin X_p. \qquad (6)$$

Шаг 3. Обновить данные: $X_p= X_p \cup\{x_j^*\}$; $A_p= A_p \cup (x_i,x_j^*)$. Присвоить $\beta(x_j^*)=1$.

Шаг 4. Если $|X_p|=n$, то остановиться. Ребра в A_p образуют кратчайший остов графа.

Если $|X_p|<n$, то перейти к шагу 2.

2.4 КОНТРОЛЬНЫЙ ПРИМЕР

Для примера рассмотрим граф, изображенный на рисунке 12. Найдем для него кратчайшее остовое дерево, используя для этой цели рассмотренный выше алгоритм Прима-Краскала. Обозначим множество смежных вершин, не входящих в порожденное поддерево, как $\Gamma^*(X_p)$. Таким образом, для всех вершин, входящих в это множество, оценка $\beta(x_j)=0$, $\forall x_j \in \Gamma^*(X_p)$. Вектор B представляет собой множество оценок $\beta(x_i)$ для всех вершин графа G: $\forall x_i \in X$. Длина порожденного поддерева обозначается как L.

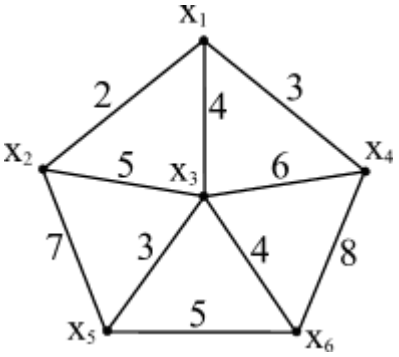


Рисунок 12.

Итерация 1: $X_p=\{x_1\}$; $A_p=\varnothing$; $\Gamma^*(X_p)=\{x_2, x_3, x_4\}$;
$c(x_1, x_2^*)=2$; $B=\{1,1,0,0,0,0\}$; $L=2$.

Итерация 2: $X_p=\{x_1, x_2\}$; $A_p=\{(x_1, x_2)\}$;
$\Gamma^*(X_p)=\{x_3, x_4, x_5\}$; $c(x_1,x_4^*)=3$;
$B=\{1,1,0,1,0,0\}$; $L=2+3=5$.

Итерация 3: $X_p=\{x_1, x_2, x_4\}$; $A_p=\{(x_1, x_2); (x_1, x_4)\}$;
$\Gamma^*(X_p)=\{x_3, x_5, x_6\}$; $c(x_1,x_3^*)=4$;
$B=\{1,1,1,1,0,0\}$; $L=5+4$.

Итерация 4: $X_p=\{x_1, x_2, x_3, x_4\}$; $A_p=\{(x_1, x_2); (x_1, x_4); (x_1,x_3)\}$;
$\Gamma^*(X_p)=\{x_5, x_6\}$; $c(x_3, x_5^*)=2$; $B=\{1,1,1,1,1,0\}$; $L=9+3=12$.

Итерация 5: $X_p=\{x_1, x_2, x_3, x_4,x_5\}$; $A_p=\{(x_1, x_2); (x_1, x_4); (x_1,x_3); (x_3,x_5)\}$;
$\Gamma^*(X_p)=\{x_6\}$; $c(x_3,x_6^*)=4$; $B=\{1,1,1,1,1,1\}$; $L=12+4=16$.

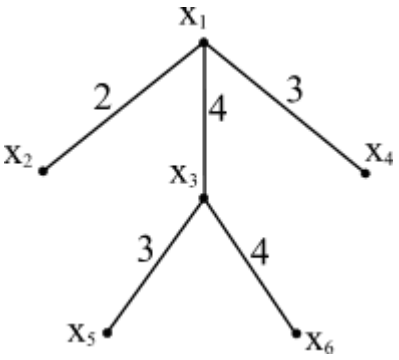


Рисунок 13.

Задача решена. Полученные множества вершин X_p и ребер A_p составляют кратчайшее остовое дерево:

$X_p=\{x_1, x_2, x_3, x_4,x_5,x_6\}$;

$A_p=\{(x_1, x_2); (x_1, x_4); (x_1,x_3); (x_3,x_5); (x_3,x_6)\}$;

Суммарная длина кратчайшего остового дерева $L=16$.

Результат решения задачи представлен на рисунке 13.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Получить задание у преподавателя в виде исходного неориентированного графа:

3.2 Составить блок-схему программы, определяющей кратчайшее остовое дерево графа с помощью алгоритма Прима-Краскала.

3.3 Создать программу, реализующую алгоритм Прима-Краскала. Исходный граф задается в виде матрицы смежности, вводимой построчно с помощью консоли. Программа должна вывести список ребер, входящих в кратчайшее остовое дерево.

4 СОДЕРЖАНИЕ ОТЧЕТА ПО РАБОТЕ

4.1 Исходное задание и цель работы.
4.2 Блок-схема программы по п.3.2.
4.3 Распечатка текста программы по п.3.3.
4.4 Контрольный пример и результаты машинного расчета.
4.5 Выводы по работе.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Дайте определение дерева; ориентированного дерева.
5.2 Какое дерево называется остовым?
5.3 Свойства остовых деревьев. Теорема Кэли.
5.4 Что называется корнем дерева?
5.5 Как преобразовать неориентированное дерево в ориентированное?
5.6 Сколько ребер содержит остовое дерево графа?
5.7 Задача нахождения кратчайшего остова графа.
5.8 Приведите практические примеры нахождения кратчайшего остова графа.
5.9 Реализация алгоритма Прима-Краскала для нахождения кратчайшего остова графа.

ЛАБОРАТОРНАЯ РАБОТА №4

РАСКРАСКА ГРАФА

1 ЦЕЛЬ РАБОТЫ

Целью работы является изучение способа правильной раскраски графа на основе эвристического алгоритма.

2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Разнообразные задачи, возникающие при планировании производства, составлении графиков осмотра, хранении и транспортировке товаров и т.д., могут быть представлены часто как задачи теории графов, тесно связанные с так называемой "задачей раскраски". Графы, рассматриваемые в данной лабораторной работе, являются неориентированными и не имеют петель.

Граф G называют *r-хроматическим*, если его вершины *могут быть раскрашены* с использованием r цветов (красок) так, что не найдется двух смежных вершин одного цвета. Наименьшее число r, такое, что граф G является *r-хроматическим*, называется *хроматическим числом* графа G и обозначается $\gamma(G)$. Задача нахождения хроматического числа графа называется *задачей о раскраске* (или *задачей раскраски*) графа. Соответствующая этому числу раскраска вершин разбивает множество вершин графа на r подмножеств, каждое из которых содержит вершины одного цвета. Эти множества являются

независимыми, поскольку в пределах одного множества нет двух смежных вершин.

Задача нахождения хроматического числа произвольного графа явилась предметом многих исследований в конце XIX и в XX столетии. По этому вопросу получено много интересных результатов.

Хроматическое число графа нельзя найти, зная только числа вершин и ребер графа. Недостаточно также знать степень каждой вершины, чтобы вычислить хроматическое число графа. При известных величинах n (число вершин), m (число ребер) и $deg(x_1),...,deg(x_n)$ (степени вершин графа) можно получить только верхнюю и нижнюю оценки для хроматического числа графа.

Пример раскраски графа приведен на рисунке 14. Этот граф является одной из запрещенных фигур, используемых для определения планарности. Цифрами “1” и “2” обозначены цвета вершин.

Максимальное число независимых вершин графа $\alpha(G)$, равное мощности наибольшего множества попарно несмежных вершин, совпадает также с мощностью наибольшего множества вершин в G, которые могут быть окрашены в один цвет, следовательно:

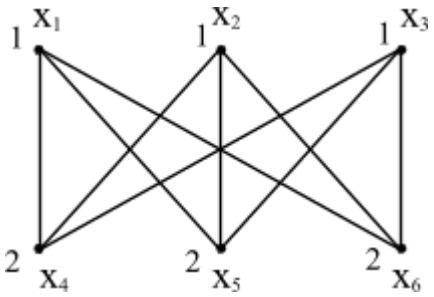


Рисунок 14 – Двудольный бихроматический граф Кенига.

$$\gamma(G) \geq \left\lceil \frac{n}{\alpha(G)} \right\rceil, \qquad (7)$$

где n - число вершин графа G, а $\lceil x \rceil$ обозначает наибольшее целое число, не превосходящее x.

Еще одна нижняя оценка для $\gamma(G)$ может быть получена следующим образом:

$$\gamma(G) \geq \frac{n^2}{n^2 - 2m}. \qquad (8)$$

Верхняя оценка хроматического числа может быть вычислена по формуле:

$$\gamma(G) \leq 1 + \max_{x_i \in X}[d(x_i) + 1]. \qquad (9)$$

Применение оценок для хроматического числа значительно сужает границы решения. Для определения оценки хроматического числа также могут использоваться другие топологические характеристики графа, например, свойство планарности.

Граф, который можно изобразить на плоскости так, что никакие два его ребра не пересекаются между собой, называется *планарным*.

Теорема о пяти красках. Каждый планарный граф можно раскрасить с помощью пяти цветов так, что любые две смежные вершины будут окрашены в разные цвета, т. е. если граф G - планарный, то $\gamma(G) \leq 5$.

Гипотеза о четырех красках (недоказанная). Каждый планарный граф можно раскрасить с помощью четырех цветов так, что любые две смежные

вершины будут окрашены в разные цвета, т. е. $\gamma(G) \le 4$, если граф G - планарный.

В 1852 г. о гипотезе четырех красок говорилось в переписке Огюста де Моргана с сэром Вильямом Гамильтоном. С тех пор эта "теорема" стала, наряду с теоремой Ферма, одной из самых знаменитых нерешенных задач в математике.

Полный граф G_n всегда раскрашивается в n цветов, равных количеству его вершин.

2.2 ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ РАСКРАШИВАНИЯ

Точные методы раскраски графа сложны для программной реализации. Однако существует много эвристических процедур раскрашивания, позволяющих находить хорошие приближения для определения хроматического числа графа. Такие процедуры также могут с успехом использоваться при раскраске графов с большим числом вершин, где применение точных методов не оправдано ввиду высокой трудоемкости вычислений.

Из эвристических процедур раскраски следует отметить последовательные методы, основанные на упорядочивании множества вершин.

В одном из простейших методов вершины вначале располагаются в порядке убывания их степеней. Первая вершина окрашивается в цвет 1; затем список вершин просматривается по убыванию степеней и в цвет 1 окрашивается каждая вершина, которая не является смежной с вершинами, окрашенными в тот же цвет. Потом возвращаемся к первой в списке неокрашенной вершине, окрашиваем ее в цвет 2 и снова просматриваем список вершин сверху вниз, окрашивая в цвет 2 любую неокрашенную вершину, которая не соединена ребром с другой, уже окрашенной в цвет 2 вершиной. Аналогично действуем с цветами 3, 4 и т. д., пока не будут окрашены все вершины. Число использованных цветов будет тогда приближенным значением хроматического числа графа.

Эвристический алгоритм раскраски вершин графа имеет следующий вид:

Шаг 1. Сортировать вершины графа по степеням убывания:
$deg(x_i) \ge deg(x_j)$, $\forall x_i, x_j \in G$; Установить текущий цвет $p=1;\ i=1.$

Шаг 2. Выбрать очередную не раскрашенную вершину из списка и назначить ей новый цвет: $col(x_i)=p$; $X_p=\{x_i\}$.

Шаг 3. $i=i+1.$ Выбрать очередную не раскрашенную вершину x_i и проверить условие смежности: $x_i \cap \Gamma(X_p)=\varnothing$, где X_p – множество вершин, уже раскрашенных в цвет $p.$ Если вершина x_i не является смежной с данными вершинами, то также присвоить ей цвет p: $col(x_i)=p.$

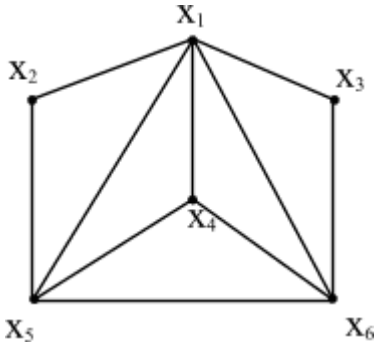
Шаг 4. Повторить п.3, пока не будет достигнут конец списка ($i=n$).

Шаг 5. Если все вершины графа раскрашены, то – конец алгоритма;
Иначе: $p=p+1;\ i=1.$ Повторить п.2.

2.4 КОНТРОЛЬНЫЙ ПРИМЕР

Раскрасим граф G, изображенный на рисунке 15. Промежуточные данные для решения задачи будем записывать в таблицу. Отсортируем вершины графа по убыванию их степеней. В результате получается вектор отсортированных вершин $X^*=\{x_1, x_5, x_6, x_4, x_2, x_3\}$.

Соответствующие данным вершинам степени образуют второй вектор: $D=\{5, 4, 4, 3, 2, 2\}$. В первой строке таблицы запишем вектор X^*; во второй – вектор D. Последующие строки отражают содержание вектора раскраски $col(X^*)$



Номера вершин X^*	x_1	x_5	x_6	x_4	x_2	x_3
Степени вершин D	5	4	4	3	2	2
$p=1$	1	-	-	-	-	-
$p=2$	1	2	-	-	-	2
$p=3$	1	2	3	-	3	2
$p=4$	1	2	3	4	3	2

Рисунок 14.

Таким образом, данный граф можно раскрасить не менее чем в четыре цвета, т.е. $\gamma(G)=4$.

3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Получить задание у преподавателя в виде исходного неориентированного графа:

3.2 Составить блок-схему программы, определяющей раскраску графа с помощью эвристического алгоритма, указанного в п.2.2.

3.3 Создать программу, реализующую эвристический алгоритм раскраски графа. Исходный граф задается в виде матрицы смежности, вводимой построчно с помощью консоли. Программа должна вывести список полученных цветов для всех вершин графа.

4 СОДЕРЖАНИЕ ОТЧЕТА ПО РАБОТЕ

4.1 Исходное задание и цель работы.
4.2 Блок-схема программы по п.3.2.
4.3 Распечатка текста программы по п.3.3.
4.4 Контрольный пример и результаты машинного расчета.
4.5 Выводы по работе.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Сформулируйте задачу раскраски графа.

5.2 Какой граф называется r-хроматическим?
5.3 Что называется хроматическим числом графа?
5.4 Как определяются нижняя и верхняя оценки хроматического числа?
5.5 Какой граф называется планарным? Во сколько цветов можно его раскрасить?
5.6 Во сколько цветов можно раскрасить полный граф?
5.7 Эвристический алгоритм раскраски графа.
5.9 Реализация эвристического алгоритма для нахождения хроматического числа графа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Н. Кристофидес. Теория графов. Алгоритмический подход. – М.: Изд. Мир, 1978, 432 с.
2. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования / Рублинецкий В.И. Введение в мир алгоритмов. – Харьков: «Фолио»; Ростов-на-Дону: «Феникс», 1997, 368 с.
3. Скворцов С.В., Хрюкин В.И. Экстремальные пути на графах: Методические указания к практическим занятиям. – Рязань: РГРТА, 1995.