

ЛАБОРАТОРНАЯ РАБОТА № 8.

“ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА”

1. ЦЕЛЬ РАБОТЫ

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

2. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

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:

$$G_p = (X_p, A_p) \subseteq G, \quad \text{где } X_p = X, A_p \subseteq A(1)$$

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 1(а), то графы на рисунках 1(б,в) являются остовами графа G . Из сформулированных выше определений вытекает, что остов графа G можно рассматривать как минимальный связанный остовый подграф графа G .

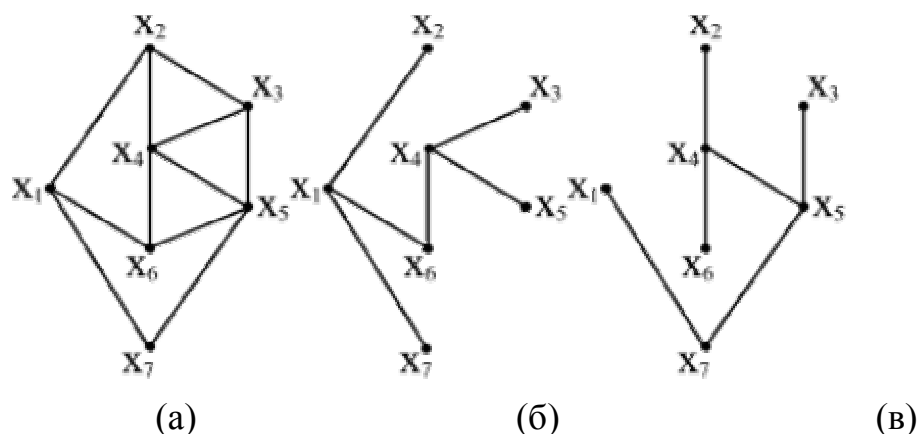


Рисунок 1 – Представление графа в виде остовых деревьев

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.

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 2 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.

Рисунок 2 –
Ориентированное
дерево

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 $A_p=\emptyset$ (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

$$c(x_i, x_j^*) = \min_{x_j \in \Gamma(X_p)} \{c(x_i, x_j)\}, \text{ где } x_i \in X_p \text{ и } x_j \notin X_p. \quad (2)$$

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

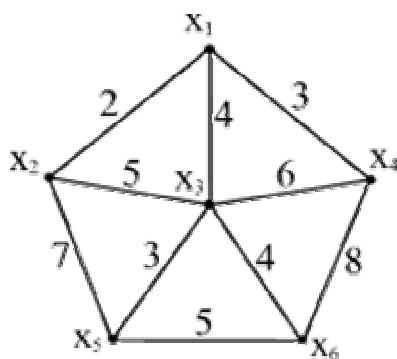


Рисунок 3 –
Пример

Для примера рассмотрим граф, изображенный на рисунке 3. 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 .

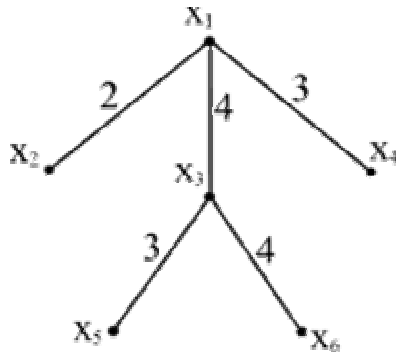
Итерация 1: $X_p=\{x_1\}$; $A_p=\emptyset$; $\Gamma^*(X_p)=\{x_2, x_3, x_4\}$; $c(x_1, x_2^*)=2$; $B=\{1, 1, 0, 0, 0, 0\}$; $L=2$.

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



Составляют кратчайшее остовое дерево:

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

Суммарная длина кратчайшего остового дерева $L = 16$.

Результат решения задачи представлен на рисунке 4.

Рисунок 4 - Результат решения

2.5 Пример программы

```

const int INF = 1000000000; // бесконечность
bool used[100] = {0}; // массив для пометок
int g[100][100] = {0}; // матрица смежности
int n; // количество вершин

```

```

intsolve() {
    int i, j, v, to;
    vector<int> min_e(n, INF), sel_e(n, -1);
    min_e[0] = 0;

    for(i = 0; i < n; i++) {
        v = -1;
        for(j = 0; j < n; j++)
            if(!used[j] && (v == -1 || min_e[j] < min_e[v]))
                v = j;
        if(min_e[v] == INF) {
            cout << "No MST!";
            exit(0);
        }
        used[v] = true;
        if(sel_e[v] != -1)
            cout << v << " " << sel_e[v] << endl;
        for(to = 0; to < n; to++)
            if(g[v][to] < min_e[to]) {

```

```

        min_e[to] = g[v][to];
        sel_e[to] = v;
    }
}
return 0;
}

```

3. ХОД РАБОТЫ

1. Получить задание у преподавателя в виде исходного неориентированного графа.

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

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. ВАРИАНТЫ ЗАДАНИЙ

Пояснения к вариантам.

1. Обозначения: «откуда», «куда» – узлы выхода и входа ребер ориентированного графа; цифры в таблице – веса ребер.

2. Для неориентированного графа элементы таблицы следует отобразить относительно главной диагонали

3. Пример ориентированного графа, соответствующего по варианту 1, дан ниже: X_i – узлы (вершины) графа, цифры на соединительных линиях – веса соответствующих ребер.

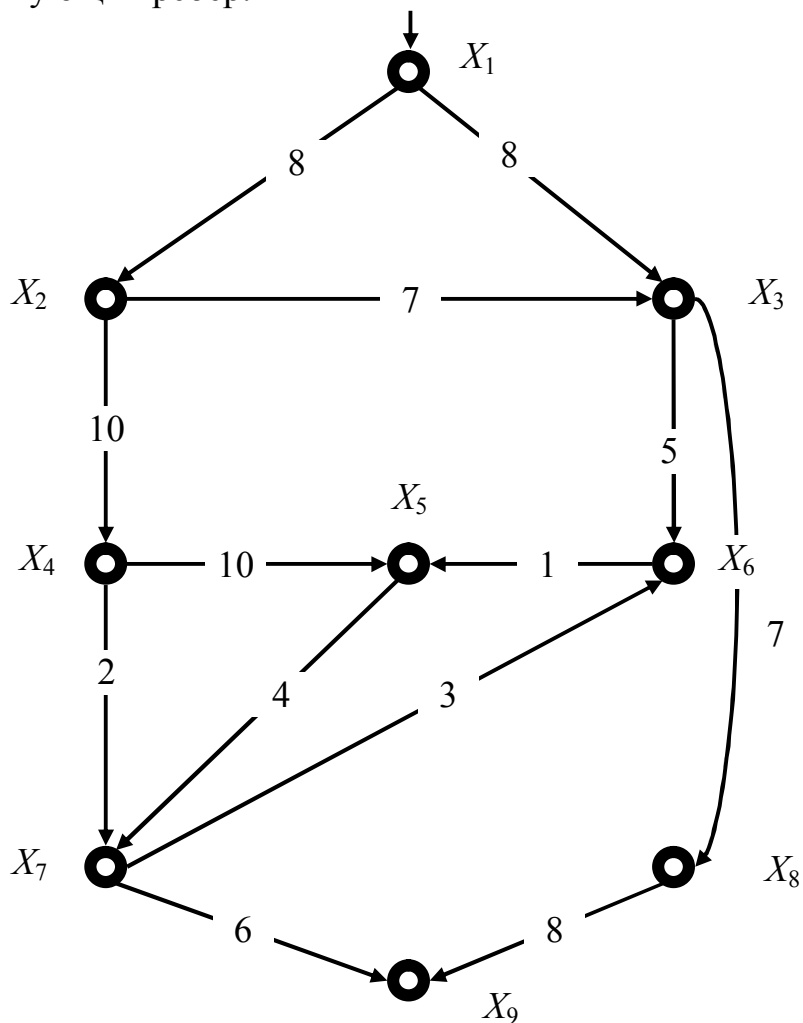


Рисунок Б.1 – Ориентированный граф к варианту 1

Вариант 7

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		9	4						
	2			7	10					
	3						4		6	
	4			6		3				
	5							1		
	6					9			7	
	7				1		8			6
	8									4
		Откуда								

Вариант 8

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		7	5						
	2				3	9				
	3		5			7	8		7	
	4					7		4		
	5							8		
	6					3				
	7						4			6
	8									8

Вариант 9

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		9	1						
	2			8	1					
	3						4		3	
	4			8		3		7		
	5							6		
	6					9			5	
	7						6		1	4
	8									1
		Откуда								

Вариант 10

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		9	8						
	2						5		1	
	3		8							
	4			9		1			4	
	5							2		
	6					8		7		
	7				2					4
	8									7

Вариант 11

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		8	6						
	2			4						
	3						5		8	
	4					6		3		
	5							10		
	6				4	2		3		
	7									9
	8									5
		Откуда								

Вариант 12

Откуда	Куда								
	1	2	3	4	5	6	7	8	9
	1	10	10						
	2			1					
	3			4		2		8	
	4				3	4			
	5						2		
	6				9				
	7			8					6
	8								9

Вариант 19

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		9	7						
	2			5	4	4				
	3						5		8	
	4						1			
	5				4		9	7		
	6								3	
	7				5		4			3
	8									6

Вариант 20

		Куда								
		1	2	3	4	5	6	7	8	9
Откуда	1		4	8						
	2			2	1	4				
	3						8	5	4	
	4					2	10			
	5							3	8	
	6					8				
	7						5			2
	8									6

5. СОДЕРЖАНИЕ ОТЧЁТА

1. Исходное задание и цель работы.
2. Структурная схема программы по п.2.
3. Распечатка текста программы по п.3.
4. Контрольный пример и результаты машинного расчета.
5. Выводы по работе.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение дерева; ориентированного дерева.
2. Какое дерево называется остовым?
3. Свойства остовых деревьев. Теорема Кэли.
4. Что называется корнем дерева?
5. Как преобразовать неориентированное дерево в ориентированное?
6. Сколько ребер содержит остовое дерево графа?
7. Задача нахождения кратчайшего остова графа.
8. Приведите практические примеры нахождения кратчайшего остова графа.
9. Реализация алгоритма Прима-Краскала для нахождения кратчайшего остова графа.