|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** |
|  |
| Институт информационных технологий и управления в технических системах |
| (полное название института) |
|  |
| кафедра «Информационные системы» |
| (полное название кафедры) |

**ОТЧЁТ**

по лабораторной работе №8

на тему

«ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА»

Выполнил

студент ИИТУТС

группы ИС/б-18-2-о

Радыгина Екатерина

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | |
|  | (должность, учёная степень преподавателя) | | | | | | | | |
|  |  | | | | | | | | |
|  | (ФИО преподавателя) | | | | | | | | |
|  | « |  | » |  |  |  | 20 | 20 | г. |
|  |  | | | | | | | | |
|  | (оценка) | | | | | | | | |
|  |  | | | | | | | | |
|  |  | | | | | | | | |

Севастополь 2020

**8.1 Цель работы**

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

**8.2 Вариант задания**

Вариант 16

Исходный граф (рисунок 8.1)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 |  | 2 | 7 |  |  |  |  |  |  |
| 1 | 2 |  |  | 8 | 3 |  |  |  |  |
| 2 | 7 |  |  |  |  |  |  | 5 |  |
| 3 |  | 8 |  |  | 4 | 1 | 3 |  |  |
| 4 |  | 3 |  | 4 |  | 6 | 10 | 6 |  |
| 5 |  |  |  | 1 | 6 |  | 6 |  |  |
| 6 |  |  |  | 3 | 10 | 6 |  |  | 9 |
| 7 |  |  | 5 |  | 6 |  |  |  | 4 |
| 8 |  |  |  |  |  |  | 9 | 4 |  |

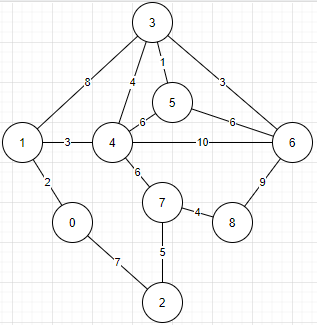


Рисунок 8.1 – Исходный граф

**8.3 Ход работы**

8.3.1 Составим схему алгоритма программы, определяющей кратчайшее остовое дерево графа с помощью алгоритма Прима-Краскала (рисунок 8.2).

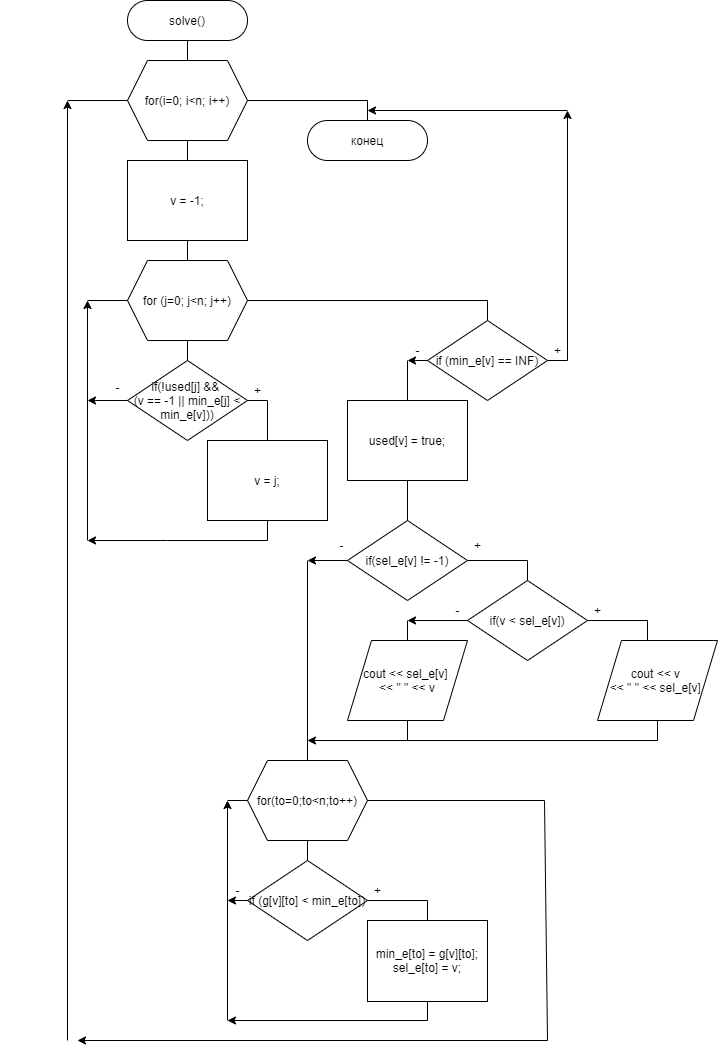


Рисунок 8.2 – Схема алгоритма Прима-Краскала

8.3.2 Создадим программу, реализующую алгоритм Прима-Краскала. Исходный граф зададим в виде матрицы смежности, вводимой построчно с помощью консоли. Программа выведет список ребер, входящих в кратчайшее остовое дерево.

Текст программы.

#include <vector>

#include <iostream>

using namespace std;

const int INF = 1000000000;

bool used[100] = {0};

int g[100][100] = {0};

int n = 9;

void solve() {

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!";

return;

}

used[v] = true;

if(sel\_e[v] != -1) {

if(v < sel\_e[v]) cout << v << " " << sel\_e[v] << endl;

else cout << sel\_e[v] << " " << 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;

}

int main() {

cout << "Input martix " << n << "x" << n << ":" << endl;

for(int i=0;i<n;i++) {

for(int j=0;j<n;j++) {

cin >> g[i][j];

if(g[i][j] == 0) g[i][j] = INF;

}

}

cout << "Edges to keep:" << endl;

solve();

return 0;

}

На рисунке 8.3 представлен пример выполнения программы.

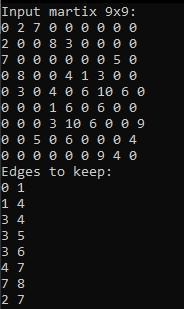


Рисунок 8.3 – Контрольный пример

На рисунке 8.4 представлено кратчайшее остовое дерево.

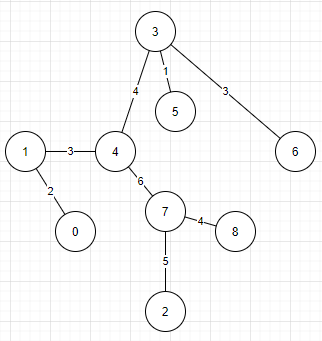


Рисунок 8.4 – Остовое дерево

**Вывод**

В ходе лабораторной работы был изучен метод построения кратчайших остовых деревьев графа на примере алгоритма Прима-Краскала.