**8 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.

**3.2 Вариант задания ­– 14**

Создать программу, реализующую алгоритм Прима-Краскала. Исходный граф (Таблица 8.1) задается в виде матрицы смежности, вводимой построчно с помощью консоли. Программа должна вывести список ребер, входящих в кратчайшее остовое дерево.

Таблица 8.1 – Граф по варианту представленный в виде таблицы

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

**3.2** **Ход работы**

С помощью таблицы 8.1 был создан граф, представленный на рисунке 8.1.

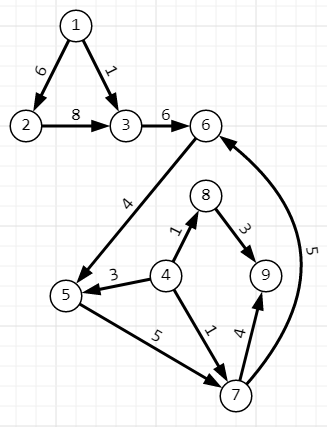


Рисунок 8.1 – Граф заданный по варианту

Была написана программа на языке С++ выполняющая требуемые задания:

#include <vector>

#include <iostream>

using namespace std;

const int INF = 1000000000;

bool used[100] = {0};

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

int n;

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(int argc,char \*\*argv) {

cout << "Input vertex count: ";

cin >> n;

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 8.2 представлено кратчайшее островое дерево, построенное по результатам работы написанной программы.

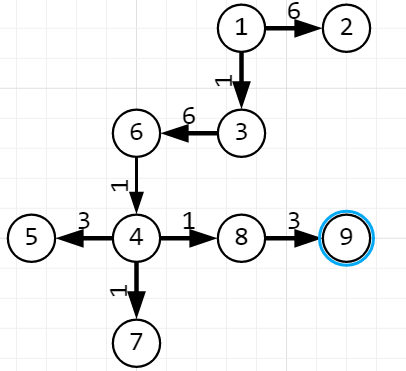


Рисунок 8.2 – Результат работы программы

**Выводы**

В ходе лабораторной работы был изучен метод построения кратчайших остовых деревьев графа на примере алгоритма Прима-Краскала. Написана программа, реализующая этот алгоритм. Для исходного графа, с помощью написанной программы, построено кратчайшее остовое дерево.