Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 8

«Построение кратчайших остовых деревьев графа»

Выполнил: ст. гр. И12д

Серегин А.В.

Проверил:

ст.пр. Шишкевич В.Е.

Севастополь

2015

1. Цель работы

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

2. Варианты задания

Реализовать алгоритм поиска кратчайшего остового дерева графа.

3. Текст программы

#include <iostream>

#include <vector>

using namespace std;

void msInit(vector<vector<int>>& g, const int n)

{

g.clear();

g.resize(n);

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

g[i] = vector<int>(n,INT\_MAX);

}

void pushArc(vector<vector<int>>& g, const int first, const int second, const int weight)

{

g[first][second] = weight;

g[second][first] = weight;

}

void solve(const vector<vector<int>> g, vector<pair<int, int>>& arcs, const int n)

{

int i, j, v, to;

vector <bool> used(n, false);

vector <int> min\_e(n, INT\_MAX);

vector <int> sel\_e(n, -1);

pair<int, int > arc;

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] == INT\_MAX)

return;

used[v] = true;

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

{

arc.first = v;

arc.second = sel\_e[v];

arcs.push\_back(arc);

}

for (to = 0; to < n; to++)

if (g[v][to] < min\_e[to])

{

min\_e[to] = g[v][to];

sel\_e[to] = v;

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

vector<vector<int>> g;

vector<pair<int, int>> arcs;

int n;

cout << "Введите кол-во узлов:" << endl;

cin >> n;

msInit(g, n);

{

cout << "Введите кол-во дуг:" << endl;

int n, first, second, weight;

cin >> n;

cout << "Введиете дуги:" << endl;

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

{

cin >> first >> second >> weight;

pushArc(g, first-1, second-1, weight);

}

}

solve(g, arcs, n);

cout << "Список дуг остового дерева:" << endl;

for (auto a : arcs)

cout << a.first+1 << ' ' << a.second+1 << endl;

system("pause");

return 0;

}

4. Результаты выполнения

На рисунке ниже изображен результат тестового примера работы программы.

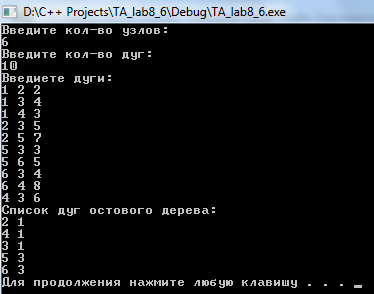


Рисунок 1 – Тестовый пример.

Выводы

В ходе лабораторной работы мы изучили алгоритм Прима-Краскала, благодаря которому мы смогли реализовать программу по нахождению минимального остового дерева графа.