# ЛАБОРАТОРНАЯ РАБОТА №6

# ИССЛЕДОВАНИЕ АЛГОРИТМА ПОСТРОЕНИЯ КРАТЧАЙШИХ

# ОСТОВЫХ ДЕРЕВЬЕВ ГРАФА

Цель работы

Исследование алгоритма Прима-Краскала построения кратчайших остовых деревьев графа методом пооператорного анализа. Получение навыков выполнения пооператорного анализа фундаментальных алгоритмов.

Задания

1. Ознакомиться с ориентированным графом, заданным по варианту (Вариант 12). Использовать данные матрицы (Рисунок 1), построив на её основании симметричную матрицу для неориентированного графа;

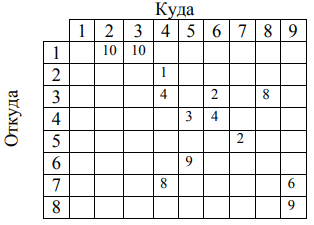


Рисунок 1 – Матрица смежности ориентированного графа

1. Написать программу, реализующую алгоритм Прима-Краскала, определяющего кратчайшее остовое дерево графа. Дополнить её операторами подсчёта количества элементарных операций;
2. Выполнить прогон программы для заданного графа. Сравнить результаты: фактическое количество операций и расчёты по функции трудоёмкости;

Текст программы

#include <iostream>

#include <vector>

using namespace std;

const int INF = 1000000000; // бесконечность

const int n = 9; // количество вершин

int g[n][n] = { // матрица смежности

{0, 10, 10, 0, 0, 0, 0, 0, 0}, // 1

{10, 0, 0, 1, 0, 0, 0, 0, 0}, // 2

{10, 0, 0, 4, 0, 2, 0, 8, 0}, // 3

{0, 1, 4, 0, 3, 4, 8, 0, 0}, // 4

{0, 0, 0, 3, 0, 9, 2, 0, 0}, // 5

{0, 0, 2, 4, 9, 0, 0, 0, 0}, // 6

{0, 0, 0, 8, 2, 0, 0, 0, 6}, // 7

{0, 0, 8, 0, 0, 0, 0, 0, 9}, // 8

{0, 0, 0, 0, 0, 0, 6, 9, 0} // 9

};

// алгоритм Прима-Краскала

void solve()

{

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

bool used[n] = { 0 }; // массив для пометок

min\_e[0] = 0;

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

{

int v = -1;

for (int 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)

cout << v + 1 << " " << sel\_e[v] + 1 << endl;

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

if ((g[v][to] < min\_e[to]) && (g[v][to] != 0))

{

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

sel\_e[to] = v;

}

}

return;

}

// Построение остового дерева графа

int main()

{

cout << "Program started: \n";

solve();

return 0;

}

Ход работы

Указанная в варианте матрица смежности была дополнена так, чтобы из ориентированного графа получить неориентированный (Рисунок 2).

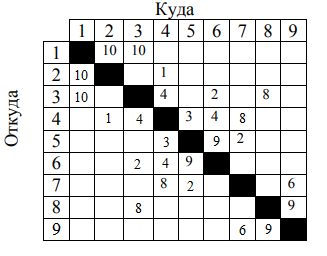


Рисунок 2 – Матрица смежности неориентированного графа

По данной матрице был графически построен заданный по варианту граф (Рисунок 3).

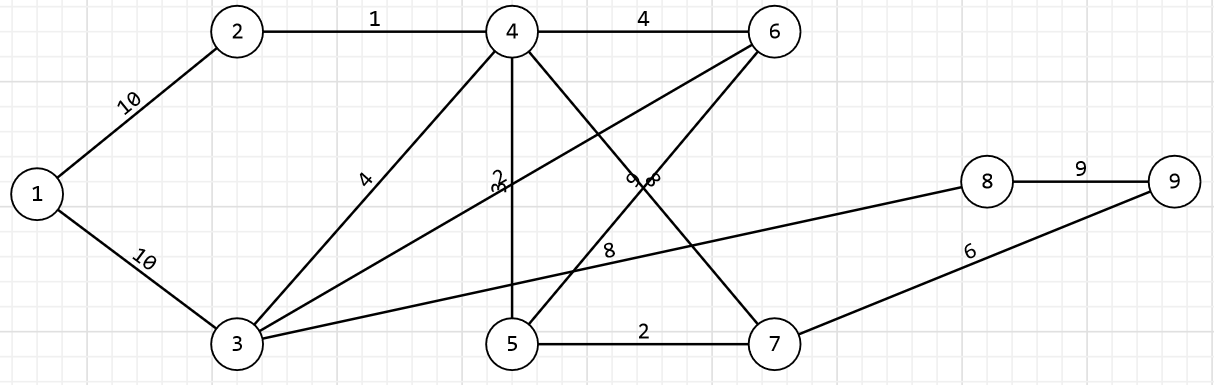


Рисунок 3 – Изображение графа

Для данного графа была запущена программа, строящая кратчайшее остовое дерево графа с помощью алгоритма Прима-Краскала. В результате на экран были выведены все рёбра, составляющее кратчайшее остовое дерево (Рисунок 4).

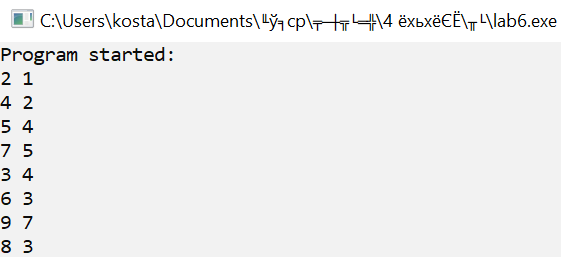


Рисунок 4 – Результат выполнения программы

Тот же метод был использован на графическом изображении графа (Рисунок 5). В результате полученные рёбра совпали с теми, что предложила программа.

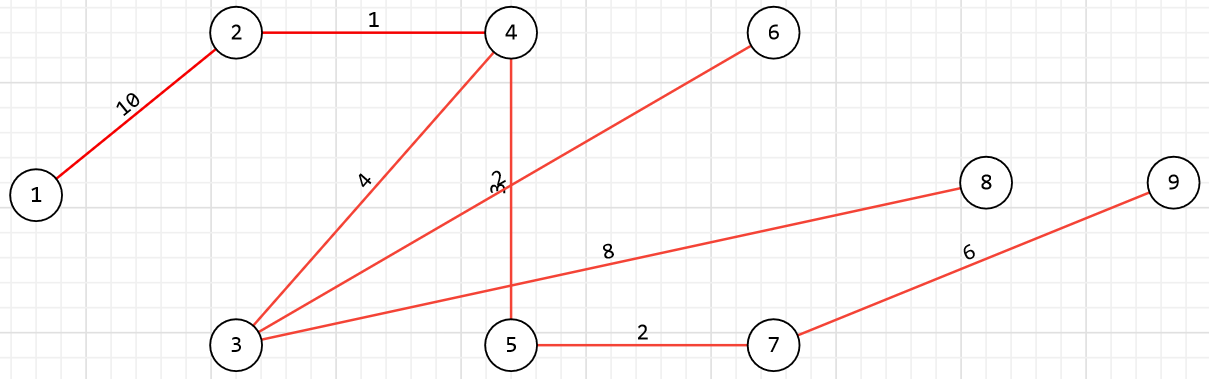


Рисунок 5 – Изображение остового дерева графа

Вывод

В ходе работы были изучены свойства остового дерева графа. Освоен алгоритм Прима-Краскала нахождения кратчайшего остового дерева графа. Написана программа, реализующая его. Данные, полученные с помощью программы, совпали с данными, полученными аналитически.