Министерствонауки и высшего образования РФ

Севастопольский государственный университет

Кафедра информатики и управления в технических системах

ОТЧЕТ

по лабораторной работе №8

ПОСТРОЕНИЕ КРАТЧАЙШИХ ОСТОВНЫХ ДЕРЕВЬЕВ ГРАФА

по дисциплине «Основы теории алгоритмов»

Выполнил:

Студент группы ИВТ/б 22-о

Черняев Н.Г.

Проверил:

Абрамович А.Ю.

г. Севастополь 2019

**Цель работы**

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

**Вариант №4**

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

Создать программу, реализующую алгоритмПрима-Краскала.

Исходный граф задается в виде матрицы смежности, вводимой построчно. Программа должна вывести список ребер, входящих в кратчайшее остовное дерево.

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

1

10

3

3

7

2

7

5

2

4

3

2

5

7

7

**Текстпрограммы**

#include <iostream>

#include <vector>

#include <fstream>

#include <algorithm>

#include <time.h>

#include <chrono>

#include <set>

using namespace std;

vector< vector<int>> graph = {

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

{ 1, 0, 3, 3, 0, 0, 7, 0, 0 },

{ 10, 3, 0, 2, 2, 7, 0, 5, 0 },

{ 0, 3, 2, 0, 4, 0, 0, 0, 0 },

{ 0, 0, 2, 4, 0, 2, 3, 0, 0 },

{ 0, 0, 7, 0, 2, 0, 5, 0, 0 },

{ 0, 7, 0, 0, 3, 5, 0, 0, 7 },

{ 0, 0, 5, 0, 0, 0, 0, 0,7 },

{ 0, 0, 0, 0, 0, 0, 7, 7, 0 }

};

set<int> SP = {}, SM = {0,1,2,3,4,5,6,7,8};

void solve ()

{

int min = numeric\_limits<int>::max();

int l, t;

for (auto i = 0; i < graph.size(); ++i)

for (auto j = i; j < graph.size(); ++j)

if (graph[i][j] < min && graph[i][j]!=0)

{

min = graph[i][j];

l = i;

t = j;

}

cout << "[" << l + 1 << ", " << t + 1 << " ]; ";

SP.insert(l); SM.erase(l);

SP.insert(t); SM.erase(t);

while (!SM.empty())

{

min = numeric\_limits<int>::max();

l = t = 0;

for (auto i:SM)

for(auto j: SP)

if (graph[i][j]<min && graph[i][j] != 0)

{

min = graph[i][j];

l = i;

t = j;

}

SP.insert(l); SM.erase(l);

cout << "[" << l+1 << ", " << t+1 << " ]; ";

}

}

int

main ()

{

auto start = chrono::steady\_clock::now ();

solve ();

auto end = chrono::steady\_clock::now ();

cout << endl

<< "Time taken in microseconds: "

<< chrono::duration\_cast < chrono::microseconds > (end - start).count ()

<< endl;

return 0;

}

**Результаты**

Пример работы программы:



То есть программа строит следующее остовное дерево:

1

3

2

5

2

3

2

7

Данное дерево является минимальным, значит программа работает верно.

**Вывод**

Вданной лабораторной работе был изучен метод построения кратчайших остовных деревьев графа на примере алгоритма Прима-Краскала.