**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Информационных систем**

Курсовая РАБОТА

**по дисциплине «Алгоритмы и структуры данных»**

Тема: поиск минимального остова методом Краскала.

|  |  |  |
| --- | --- | --- |
| Студент гр. 1395 |  | Прокофьев Д. О. |
| Преподаватель |  | Пелевин М. С. |

Санкт-Петербург

2022

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Прокофьев Д. О. | | |
| Группа 1395 | | |
| Тема работы: поиск минимального остова методом Краскала. | | |
| Исходные данные:   * сортировка * обход графов (в глубину и в ширину) * хранение графов (списки смежности, матрицы [смежности](http://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_%D1%81%D0%BC%D0%B5%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%B8_%D0%B3%D1%80%D0%B0%D1%84%D0%B0), инцидентности) * построение системы непересекающихся множеств | | |
|  | | |
| Дата выдачи задания: | | |
| Дата сдачи реферата: | | |
| Дата защиты реферата: | | |
| Студент |  | Прокофьев Д. О. |
| Преподаватель |  | Пелевин М. С. |

**содержание**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1. | Введение | 4 |
| 1.1 | Цель работы | 4 |
| 1.2. | Задание | 4 |
| 2. | Выполнение работы | 5 |
| 2.1 | Реализация сортировки | 5 |
| 2.2  2.3  2.4 | Реализация Хранение графов  Реализация Обхода графов  Реализация систем непересекающихся множеств | 5  7 |
| 3. | Вывод и тесты | 8 |
| 3.1 | Демонстрация работы программы | 8 |
| 3.2. | Вывод о проведенной работе | 9 |
|  |  |  |
|  | Список использованных источников | 10 |

**1. введение**

**1.1. Цель работы**

Продемонстрировать знания следующих вопросов:

* сортировка
* обход графов (в глубину и в ширину)
* хранение графов (списки смежности, матрицы смежности, инцидентности)
* построение системы непересекающихся множеств

**1.2. Задание**

Реализовать алгоритм поиска минимального остова на основе алгоритма Краскала (Крускала). **2. Выполнение работы**

**2.1. Реализация Сортировки**

void sort::sortMR(ver\_sm\* mas, int kol) {

ver\_sm z;

for (int i = 0; i < kol - 1; i++) {

for (int j = 0; j < kol - i - 1; j++) {

if (mas[j].ves > mas[j + 1].ves) {

z = mas[j];

mas[j] = mas[j + 1];

mas[j + 1] = z;

}

}

}

}

В данной программе реализован метод сортировки “Пузырек”. Этот алгоритм был выбран в связи с малым размером массива, стабильностью и отсутствием требованием дополнительной памяти.

**2.2. Реализация Хранение графов**

Реализован через список смежности

struct ver\_sm {

int ver1;

int ver2;

int ves;

};

**2.3 Реализация Обхода графов**

int BFS::BFS\_cp(int\*\* matr, int kol) {

distance = new int[kol];

prevTop = new int[kol];

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

distance[i] = -1;

prevTop[i] = -1;

}

for (int start = 0; start < kol; start++) {

if (distance[start] == -1) {

distance[start] = 0;

Q.push(start);

while (Q.size()) {

int u = Q.front();

Q.pop();

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

int v = matr[u][j];

if (distance[v] == -1) {

distance[v] = distance[u] + 1;

prevTop[v] = u;

Q.push(v);

}

}

}

count++;

}

}

delete[] distance;

delete[] prevTop;

return count;

}

Данный алгоритм находит количество компонент связностей

**2.4 Реализация систем непересекающихся множеств**

for (int i = 0; i < kol\_ver; i++) {

comp[i] = i;

}

if (comp[vera[i].ver1] != comp[vera[i].ver2]) {

for (int j = 0; j < kol\_reb; j++) {

if (comp[j] == comp[vera[i].ver2]) {

comp[j] = comp[vera[i].ver1];

}

}

Реализована система непересекающихся множеств на массиве

**3. Вывод и тесты**

**3.1. Демонстрация работы программы**

input

**a b c d e f g h i j k l**

**0 2 0 0 49 0 0 0 0 0 0 0**

**0 0 7 0 0 13 0 0 0 0 0 0**

**0 0 0 22 0 0 40 43 0 0 0 0**

**0 0 0 0 0 0 0 44 0 0 0 0**

**0 0 0 0 0 41 0 0 27 0 0 0**

**0 0 0 0 0 0 38 0 16 15 29 0**

**0 0 0 0 0 0 0 24 0 0 47 50**

**0 0 0 0 0 0 0 0 0 0 0 12**

**0 0 0 0 0 0 0 0 0 9 0 0**

**0 0 0 0 0 0 0 0 0 0 10 0**

**0 0 0 0 0 0 0 0 0 0 0 19**

**0 0 0 0 0 0 0 0 0 0 0 0**

Output

a b

b c

i j

j k

h l

b f

f j

k l

c d

g h

e i

160

Программа работает корректно

**3.2. Вывод о проведенной работе**

В результате работы был реализован поиск минимального остовного дерева, а также изучены и закреплены знания по хранению и методами работы с графами.

**список использованных источников**

* Лекции Пелевина М. С.