

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Построение и анализ алгоритмов»**  
**Тема: Жадный алгоритм и  $A^*$**

Студентка гр. 7304

\_\_\_\_\_

Нгуен Т.Т. Зуен

Преподаватель

\_\_\_\_\_

Филатов А.Ю.

Санкт-Петербург

2019

## Цель работы:

Исследование жадного алгоритма и  $A^*$  и их реализация на языке C++.

## Задание:

Разработайте программу, которая решает задачу построения пути в *ориентированном* графе при помощи **жадного алгоритма**. Жадность в данном случае понимается следующим образом: на каждом шаге выбирается последняя посещённая вершина. Переместиться необходимо в ту вершину, путь до которой является самым дешёвым из последней посещённой вершины. Каждая вершина в графе имеет буквенное обозначение ("a", "b", "c"...), каждое ребро имеет неотрицательный вес.

Разработайте программу, которая решает задачу построения кратчайшего пути в *ориентированном* графе **методом  $A^*$** . Каждая вершина в графе имеет буквенное обозначение ("a", "b", "c"...), каждое ребро имеет неотрицательный вес. В качестве эвристической функции следует взять близость символов, обозначающих вершины графа, в таблице ASCII.

В первой строке через пробел указываются начальная и конечная вершины

Далее в каждой строке указываются ребра графа и их вес

В качестве выходных данных необходимо представить строку, в которой перечислены вершины, по которым необходимо пройти от начальной вершины до конечной.

## Описание Жадного алгоритма

Производится обход ориентированного графа по путям которые были заданы в входных данных.

Класс Path, содержащий информация одного ребра графа (начальная, конечная вершина и его неотрицательный вес).

Функция greedy для поиска пути от начальной вершины до конечной с минимальном весом. Поиск пути от начальной до одного из близлежащих вершин, которая имеет наименьший вес. Закончит алгоритм при переходе в конечную точку.

## Описание A\* алгоритма

Производится обход ориентированного графа по путям которые были заданы в входных данных.

Класс Vertex, содержащий информация одного вершины в графе и вершин, которых он может достичь.

Класс Path, содержащий информация одного ребра графа (начальная, конечная вершина и его неотрицательный вес).

Функция Astar для поиска пути от начальной вершины до конечной с минимальным весом. Если веса путей были равным, тогда пойдет по дороге с меньшим количеством ребер. В процессе работы алгоритма для вершин рассчитывается функция  $fScore = gScore + heuristic$ , где

$gScore$  - наименьшая стоимость пути в  $v$  из стартовой вершины,

$heuristic$  - эвристическое приближение стоимости пути от текущей до конечной цели.

## Результаты:

### 1. Жадный алгоритм:

```
a e
a b 3.0
b c 1.0
c d 1.0
a d 5.0
d e 1.0
a a a
abcde
Press any key to continue . . .
```

### 2. A\*

```
a e
a b 3.0
b c 1.0
c d 1.0
a d 5.0
d e 1.0
a a a
ade
Press any key to continue . . .
```

**Выводы:**

В результате работы программы были реализованы жадный алгоритм и A\* алгоритм нахождения пути на ориентированном графе и протестированы.