**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Построение и анализ алгоритмов»**

Тема: «Алгоритмы на графах»

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Вероха В. Н. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2018

**Цель работы:**

Изучение жадного алгоритма и алгоритма A\* для поиска кратчайших путей в графе.

**Условия задания:**

*Задание 2.1:*

Разработайте программу, которая решает задачу построения пути в 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.  
Пример входных данных:

a e  
a b 3.0  
b c 1.0  
c d 1.0  
a d 5.0

d e 1.0

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. Для приведённых в примере входных данных ответом будет:

abcde

*Задание 2.2:*

Разработайте программу, которая решает задачу построения кратчайшего пути в графе методом А\*. 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.  
Пример входных данных:

a e

a b 3.0

b c 1.0

c d 1.0

a d 5.0

d e 1.0

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. Для приведённых в примере входных данных ответом будет:

Ade

**Описание алгоритмов:**

*Жадный алгоритм*

При действии жадного алгоритма на каждом шаге выбирается ребро с минимальным весом, при этом нет гарантии, что итоговое решение будет оптимальным. Список инцидентности для заданного графа сортируется в порядке возрастания значения ребер для каждой вершины. Отсортированный список посылается в функцию поиска пути, которая берет первую вершину в списке инцидентности и проходит по вершинам. Функция продолжает работать, пока список не опустеет, т. е. будут проверены все вершины и найден путь до конечной вершины.

*Алгоритм A\**

Алгоритм A\* является аналогом жадного алгоритма, однако находит маршрут наименьшей стоимости от начальной вершины до выбранной конечной. В процессе работы алгоритма рассчитываем приоритет вершины, который складывается из текущего наименьшего пути до этой вершины и эвристического приближения стоимости пути. Открытые приоритеты вершины храним в очереди с приоритетом по значению. При стартовом состоянии в очереди находится вершина с приоритетом, заданным нулем.

**Описание функций:**

*Жадный алгоритм:*

*void searchWay(vector<char> tops, vector<vector<Node>> nodes, int startIndex, char end, stack <char> stack) –* функция делает проверку первого элемента в списке инцидентности на конечность, т. е. является ли он конечным. Если текущий элемент не конечный, то функция рекурсивно вызывается и продолжается поиск уже со следующей вершины. Если это конечный элемент, то функция завершается.

*void sortiForNodes(vector<vector<Node>> &nodes)*- функция сортировки по возрастанию значения весов ребер списка инцидентности.

*void printStack(stack <char> stack)* – функция печати пути.

*Алгоритм А\*:*

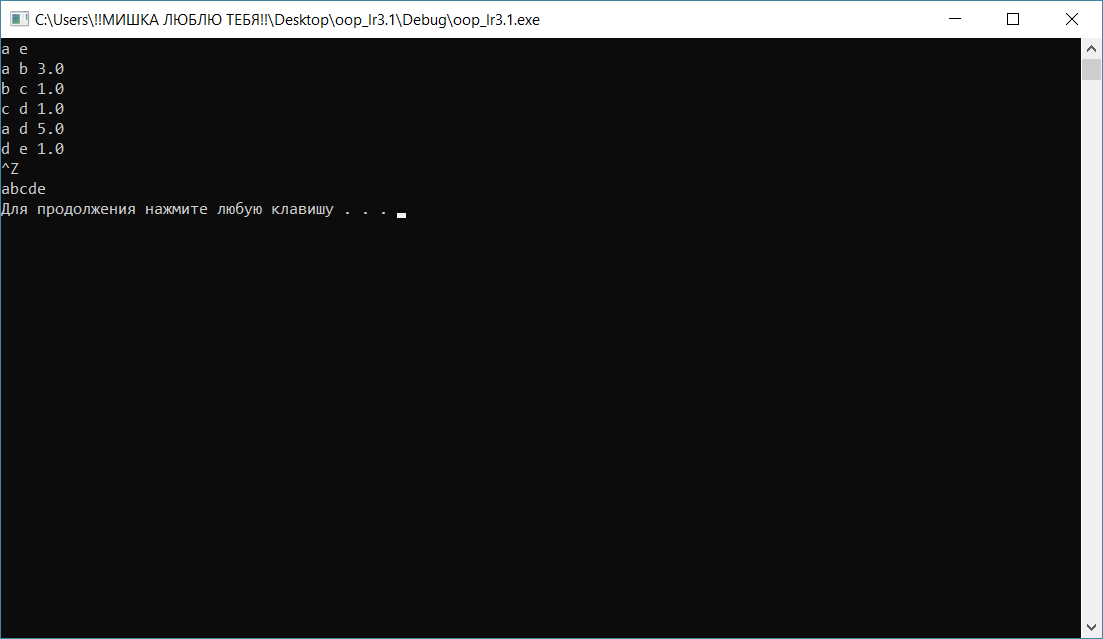
*void searchShortWay(vector<char> tops, vector<vector<Node>> nodes, char start, char end) –* при стартовом состоянии заносит в очередь с приоритетом стартовую вершину с нулевым приоритетом. Пока очередь не пуста, функция берет первый элемент очереди, находит приоритеты всех доступных вершин и заносит их в очередь. Остановка: очередь пуста, либо найден путь из начальной в конечную вершину.

*double heuristic(char top\_1, char top\_2)* - функция возвращает оцененное расстояние (эвристическое приближение) от вершины top\_1 до вершины top\_2 как разницу их номеров в таблице ASCII.

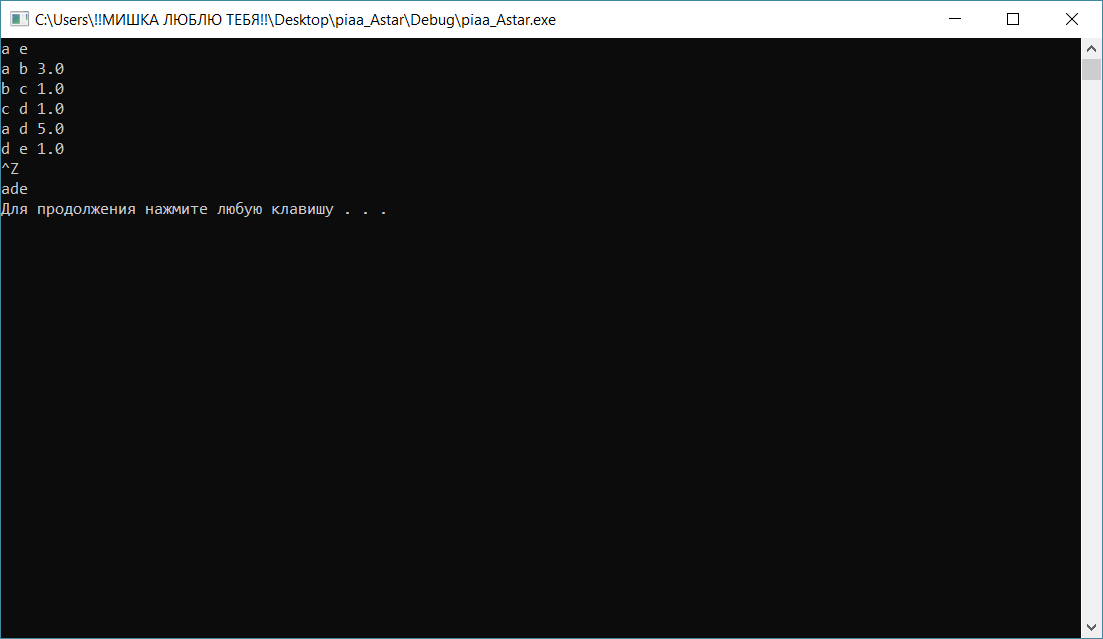
**Тестирование:**

*Тест 1*

Жадный:

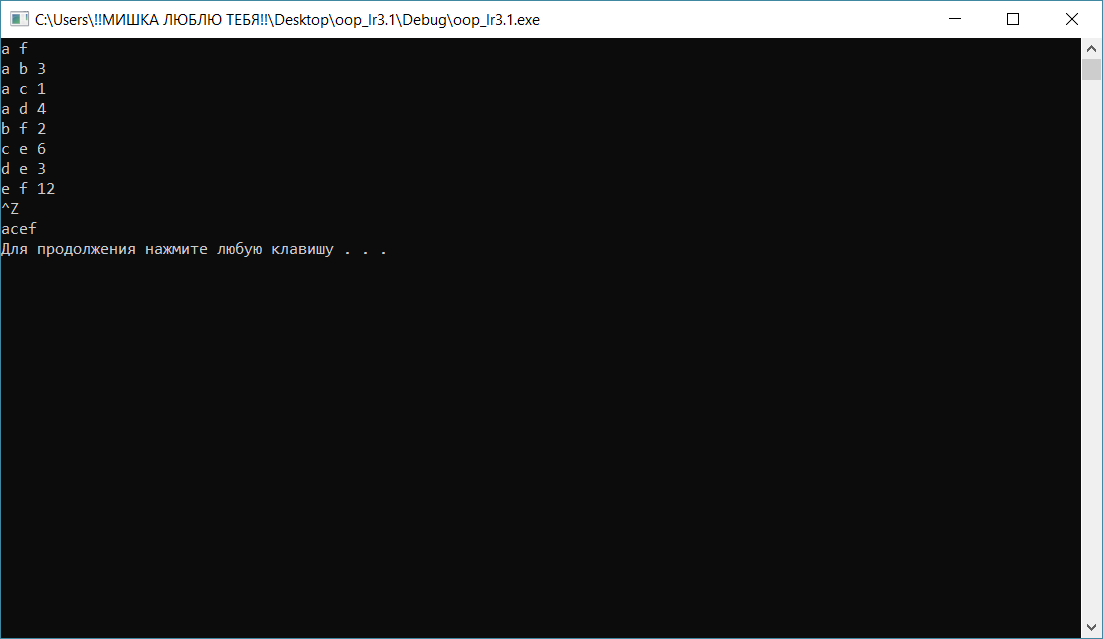


А\*:

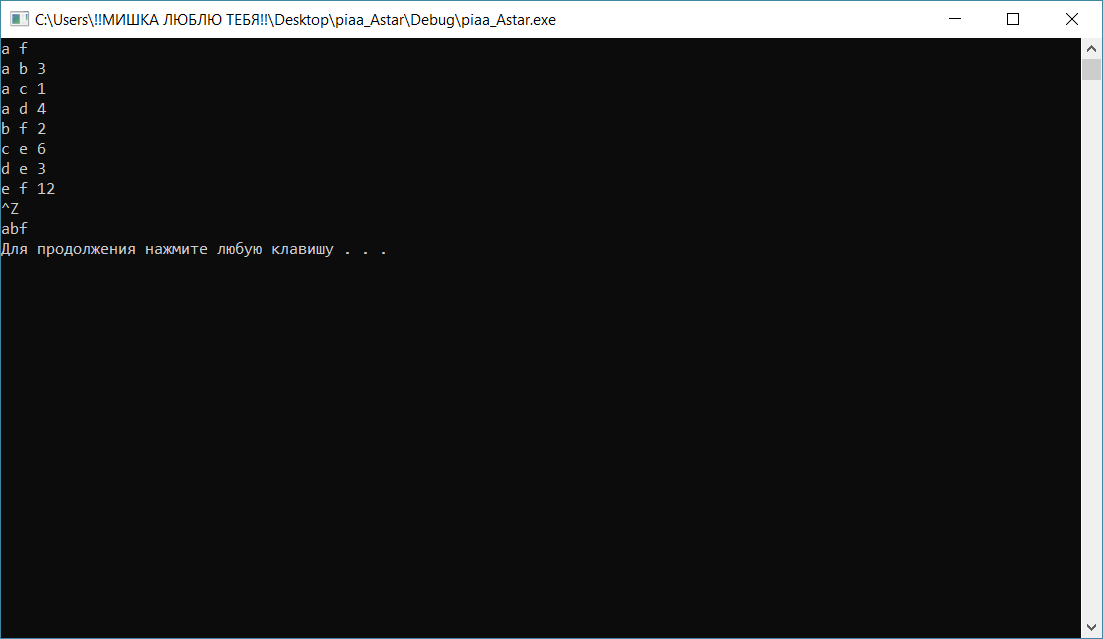


*Тест 2*

Жадный:

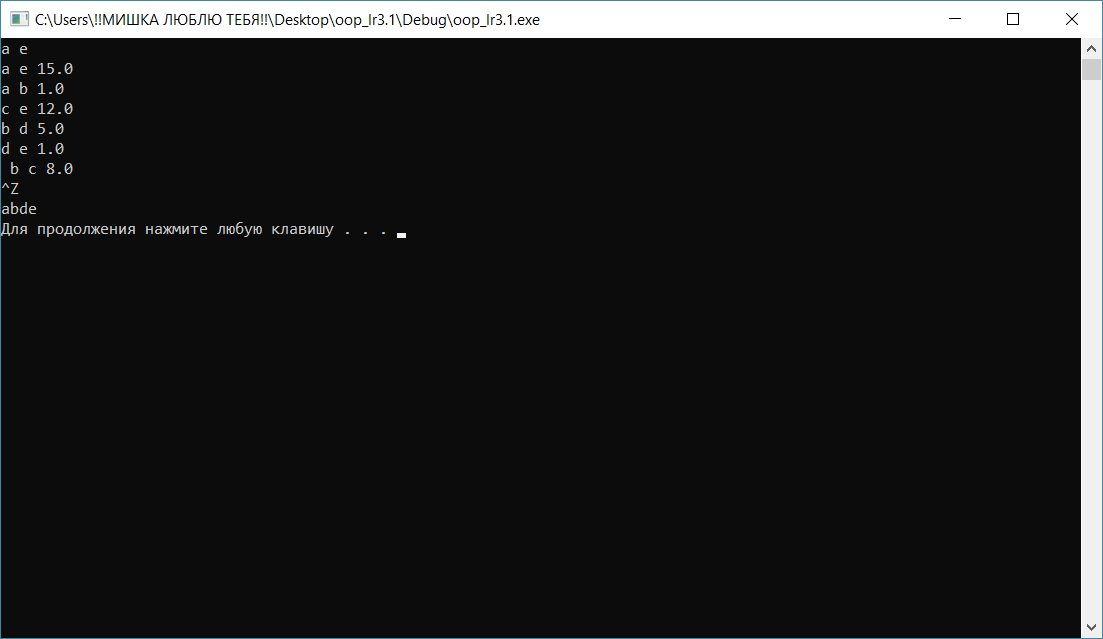


А\*:

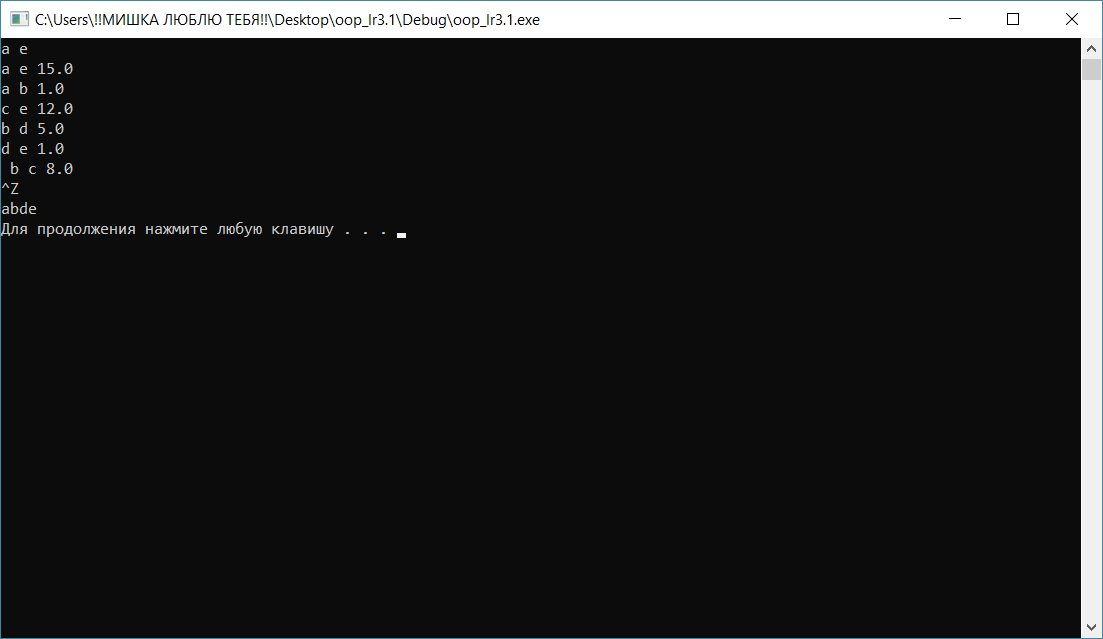


*Тест 3*

Жадный:



А\*:



**Вывод:**

В процессе выполнения лабораторной работы был реализован жадный алгоритм для поиска пути в графе и алгоритм А\* для поиска кратчайшего пути в графе.