

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: Поиск с возвратом

Студент гр. 1304

Павлов Д.Р.

Преподаватель

Шевелева А.М.

Санкт-Петербург

2023

Цель работы.

Изучить жадный алгоритм и алгоритм A*. Решить задачу построения пути в ориентированном графе с наименьшим весом ребер, используя жадный алгоритм и алгоритм A*.

Задание.

1. 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.

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.

2. 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 классов.

IReader — абстрактный класс, определяющий интерфейс для чтения исходных данных. Содержит абстрактный метод `ReadGraph()`, который должен быть реализован в наследниках.

FileReader — класс-наследник `IReader`, который реализует метод `ReadGraph()` для чтения исходных данных из файла. Читает данные, описывающие граф, из файла `input.txt` и возвращает стартовую и конечную вершины, а также представление графа в виде словаря

TerminalReader — класс-наследник `IReader`, который реализует метод `ReadGraph()` для чтения исходных данных из терминала. Читает данные, описывающие граф, из стандартного ввода и возвращает стартовую и конечную вершины, а также представление графа в виде словаря.

IAlgorithm — абстрактный класс, определяющий интерфейс для алгоритмов поиска пути в графах. Содержит абстрактный метод `GetPath()`, который должен быть реализован в наследниках.

GreedAlgorithm — класс-наследник `IAlgorithm`, который реализует алгоритм поиска пути при помощи жадного алгоритма. Использует локально оптимальные решения на каждом этапе, предполагая, что итоговое решение также будет оптимальным.

AStar — класс-наследник `IAlgorithm`, который реализует алгоритм поиска пути при помощи алгоритма A^* . Использует эвристическую функцию для расчета оценки расстояния между текущей вершиной и конечной вершиной, определяющей порядок поиска.

Выводы.

Изучена работа жадного алгоритма и алгоритма A^* . Решена задача построения пути в ориентированном графе с наименьшим весом ребер, используя жадный алгоритм и алгоритм A^* . Работа прошла как собственные тесты, так и тесты на платформе Stepik. Для решения задачи было решено использовать ООП и оба алгоритма были реализованы как методы одного класса. При написании решающая роль была отведена такому типу данных как словарь и абстрактная структура данных очередь с приоритетом. В качестве эвристической функции для алгоритма A^* была использована разность ASCII символов, обозначающих имя конечного и текущего узла.