МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный

электротехнический университет

«ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МО ЭВМ

**отчет**

по практической работе №2

по дисциплине «Построение и анализ алгоритмов»

Тема: Алгоритмы поиска пути в графе

|  |  |  |
| --- | --- | --- |
| Студент гр. 9303 |  | Дюков В. А. |
| Преподаватель |  | Шевская Н. В. |

Санкт-Петербург

2021

Цель работы

Ознакомиться с алгоритмами поиска кратчайших путей в ориентированном графе. Использовать эти алгоритмы при поиске решений для лабораторных работ.

Задание

Вариант 5: реализовать алгоритм Дейкстры поиска пути в графе (на основе кода к *A\**).

Разработайте программу, которая решает задачу построения пути в ориентированном графе при помощи **жадного алгоритма** и **методом А\***.

Для жадного алгоритма на каждом шаге выбирается последняя посещённая вершина. Переместиться необходимо в ту вершину, путь до которой является самым дешёвым из последней посещённой вершины.

В реализации метода А\*качестве эвристической функции следует взять близость символов, обозначающих вершины графа, в таблице ASCII.

Каждая вершина в графе имеет буквенное обозначение, каждое ребро имеет неотрицательный вес.

Выполнение работы

Для выполнения работы было создано пространство имён *daa\_2* в котором были определены следующие типы:

* typedef std::pair<char, char> doub\_cc;
* typedef std::map<char, char> dmap\_cc;
* typedef std::pair<float, char> doub\_fc;
* typedef std::multimap<float, char> mmap\_fc;
* typedef std::map<char, mmap\_fc> dmap\_cs;
* typedef std::map<char, float> dmap\_cf;
* typedef std::vector<char> dvec\_c\_;

и также определены следующие функции:

* bool greedy(doub\_cc arc, dmap\_cs& map, dvec\_c\_& vec);
* bool dijkstra(doub\_cc arc, dmap\_cs& map, dvec\_c\_& vec);
* bool a\_star(doub\_cc arc, dmap\_cs& map, dvec\_c\_& vec);
* template <class T>

void turn\_vect(std::vector<T>& data);

Функции *greedy*, *dijkstra*, *a\_star* производят поиск пути в графе при помощи жадного алгоритма, алгоритма Дейкстры и метода A\* соответственно. Функция *greedy* при этом является рекурсивной. Результирующую цепочку вершин графа все три метода сохраняют в аргумент *vec* – последний аргумент функций. Функции возвращают *true* в случае, если путь от стартовой до конечной вершины существует и найден, *false* – в противном случае.

Шаблонная функция *turn\_vect* меняет местами элементы вектора относительно его середины. Результат сохраняется в том же векторе, который был передан в качестве аргумента.

Также в *daa\_2* были переопределены операторы ввода и вывода для считывания графа из входного потока и вывода его в консоль, для вывода результирующего вектора и для считывания пары элементов, между которыми производится поиск пути.

Оценка сложности алгоритмов

Сложность жадного алгоритма равна в худшем случае O(E), где E – число ребер графе.

Сложность алгоритма Дейкстры равна O(V2), где V – число вершин графа.

Сложность алгоритма A\* варьируется в зависимости от устройства эвристики. В худшем случае сложность алгоритма равна сложности алгоритма Дейкстры для того же графа – O(V2). В лучшем случае (когда эвристика практически не отклоняется от оптимальной) сложность алгоритма равна O(V log(V)).

Тестирование

Тестирование проводилось на графах матричного вида. Пример такого графа представлен на рис. 1. Тесты представленны в таблице 1.

|  |
| --- |
| Изображение выглядит как электроника, клавиатура, закрыть  Автоматически созданное описание |
| Рисунок 1 – Граф таблица соответствия стоимости перехода в вершину из любой соседней и цвета вершины (справа) |

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 1 – Тестирование | | | |
| Тестируемый граф | Выходные данные | | |
| жадный алгоритм | алгоритм Дейкстра | A\* |
| Путь из ‘e’ в ‘u’  Изображение выглядит как электроника, клавиатура, закрыть  Автоматически созданное описание | edihmlqpu | edihmlqpu | ejinmrqvu |
| Путь из ‘a’ в ‘e’  Изображение выглядит как электроника, клавиатура, закрыть  Автоматически созданное описание | abgfklmnsrqpuvwxytojihcde | abcde | abcde |
| Путь из ‘a’ в ‘y’  Изображение выглядит как электроника, закат, легкий, калькулятор  Автоматически созданное описание | abcdejihgfklmnotsrqpuvwxy | abcdejoty | afkpuvwxy |

Вывод

В ходе проделанной работы был реализован жадный алгоритм, алгоритм Дейкстра, а также алгоритм А\* для решения поиска пути во взвешенном графе.