

-МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
КафедраМО ЭВМ

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

Студентгр.7304

—

Ажель И . В

Преподаватель

—

Филатов А.Ю

г. Санкт-Петербург
2019

Цель работы

Освоить жадный алгоритм, а так же алгоритм A^* , который является модификацией Дейкстры. Реализовать данные алгоритмы на языке C++.

Задача

1. Разработать 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 посещенная вершина. Переместиться необходимо в ту вершину, путь до которой является самым дешёвым из последней посещенной вершины. Каждая вершина в графе имеет буквенное обозначение ("a", "b", "c..."), каждое ребро имеет неотрицательный вес.
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.

Ход работы

1. Была разработана программа, которая решает задачу построения пути в ориентированном графе при помощи жадного алгоритма. Граф хранится в виде списка связности. Алгоритм работает следующим образом. Поиск пути начинается со стартовой вершины. Среди соседей данной вершины ищется минимальная вершина (ребро, которое имеет минимальный размер), которая еще не просмотрена. Если такая вершина была найдена, то мы переходим к этой вершине и повторяем алгоритм. Если такой вершины среди соседей нет (то есть все просмотрены, но мы не достигли конечной вершины), то мы возвращаемся на предыдущую вершину и повторяем алгоритм. Жадный алгоритм не гарантирует нахождение самого короткого пути.
2. Была разработана программа, которая решает задачу построения кратчайшего пути в ориентированном графе **методом A^*** . Граф хранится в виде списка связности. Алгоритм работает следующим образом. В данной программе есть контейнер, который хранит расстояния от стартовой вершины до данной не просмотренной вершины, но не все а лишь соседей просмотренных вершин. В начале поиска этот контейнер хранит одну вершину, которая является начальной и имеет нулевое расстояние. Среди этого контейнера ищется такая вершина, для которой расстояние от начальной является минимальное. Расстояние вычисляется, как путь до этой вершины + значение эвристической функции. Эта вершина помечается как посещенная и дальше выполняются следующие действия: начинается

просмотр не посещенных соседей данной вершины, если сосед не в данном контейнере, то вычисляется расстояние до него и он кладется в контейнер. Если сосед есть в данном контейнере, то смотрим, если перейти в соседа из данной вершины, то это расстояние будет меньше, чем расстояние в контейнере для данной вершины. Если да, то мы изменяем значение расстояния для данной вершины. Иначе идем дальше. Если в контейнер была добавлена вершина или для нее изменено расстояние, то чтобы построить обратный путь в другой контейнер кладется следующая пара $\text{pair} \langle \text{char}, \text{char} \rangle$ $\text{сосед} \neq \text{текущая}$.

Последнее действие, это удаление из контейнера текущей вершины и переход к следующему шагу. Алгоритм заканчивается, когда текущей вершиной стала конечная вершина.

Примеры работы программы

1. Пример работы жадного алгоритма

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

abcde
```

2. Пример работы A*

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

ade
```

Вывод

В ходе данной лабораторной работы были реализованы два алгоритма. Жадный алгоритм ищет путь от точки А до точки В и при этом не гарантирует нахождения самого короткого пути. Жадность алгоритма подразумевает, что если выбирать на каждом шаге наименьшее значение, то и придем мы в точку В по самому короткому пути. Алгоритм A* ищет самый короткий путь из точки А в точку В. Этот алгоритм является модификацией алгоритма Дейкстры, в него вводится эвристическая функция и в итоге расстояние вычисляется следующим образом $f(x) = g(x) + h(x)$, где x – данная вершина, $g(x)$ – это расстояние от начальной вершины до данной, а $h(x)$ – это эвристическая функция, которая является индивидуальной для каждой задачи.

