|  |
| --- |
| Санкт-петербургский государственный университет |
| Задание по курсу «Алгоритмы и анализ сложности» |
| Алгоритм Дейкстры поиск кратчайших путей в орграфе |
|  |
| **Терещенко Дмитрий Владиславович (группа 17.Б13-пу)** |
| **25.11.2019** |

|  |
| --- |
|  |

Содержание

1. Введение
2. Алгоритм
3. Математический анализ
4. Эмпирический анализ
5. Литература

I. Введение

**Алгоритм** **Дейкстры** — алгоритм на графах, изобретённый нидерландским учёным *Эдсгером Дейкстрой* в 1959 году. Решает задачу поиска кратчайших путей из одной вершины во взвешенном ориентированном графе в случае, когда веса ребер неотрицательны. Алгоритм широко применяется в программировании и технологиях, например, его используют протоколы маршрутизации [OSPF](https://ru.wikipedia.org/wiki/OSPF) и [IS-IS](https://ru.wikipedia.org/wiki/IS-IS).

Существует множество вариантов реализации данного алгоритма. Все они отличаются выбором структуры данных, но основные шаги остаются неизменны, а именно:

* Хранение доп. информации о вершине: о её посещении; о кратчайшей длине пути до неё
* Получение не посещённой вершины v’ с минимальным кратчайшим расстоянием
* Обновление расстояния до смежных вершин, к которым есть путь из вершины v’.

Поскольку в алгоритме Дейкстры для посещения всегда выбирается самая «лёгкая», или «близкая», вершина, можно утверждать, что этот алгоритм придерживается жадной стратегии. Жадные стратегии не всегда приводят к оптимальным результатам, однако, как видно из приведённой в *источнике [1]* теоремы 24.6 и следствия 24.7 из нее, алгоритм Дейкстры действительно находит кратчайшие расстояния.

II. Алгоритм

Данная реализация выбрана из *источника [2]*

# Обозначения

* ***V* —** множество вершин графа
* ***E* —** множество рёбер графа
* **—** вес (длина) ребра
* ***s*** **—** вершина, расстояния от которой ищутся
* ***v*** **—** текущая рассматриваемая вершина
* **—** равно *True*, если вершина ***u*** была посещена, иначе *False*
* **—** по окончании работы алгоритма равно длине кратчайшего пути из ***s*** до вершины ***u***

# Псевдокод

Присвоим

Для всех отличных от ***s***

присвоим

присвоим

Присвоим

Присвоим

Пока

изменим

Для всех

если (и  и ) то

изменим

изменим

Для всех

если ( и ) то

изменим

изменим

III. Математический анализ

Воспользуемся общим планом математического анализа эффективности не рекурсивных алгоритмов (*источник [4]*)*:*

1. Размер выходных данных: кол-во вершин в графе ( )
2. Основная операция алгоритма: ***сравнение***
3. Помимо размера входных данных сложность будет зависеть также от кол-ва рёбер, от их весов и от наличия связей со стартовой вершиной. Поэтому рассмотрим:

* ***Лучший случай***: вершины в графе никак не связаны со стартовой вершиной
* ***Средний случай***: не очевиден
* ***Худший случай***: полный граф

1. Подсчитаем кол-во выполняемых основных операций:

* Цикл по обновлению кратчайшего расстояния в смежных вершинах:
  + Проходит по всем вершинам => кол-во итерация равно
  + На каждой итерации проводится 3 сравнения
* Цикл по поиску не посещённой вершины с минимальным кратчайшим расстоянием:
  + Проходит по всем вершинам => кол-во итерация равно
  + На каждой итерации проводится 2 сравнения

* Основной цикл на каждой итерации делает доп. 1 сравнение
* Максимальное кол-во итераций основного цикла:
* Для выхода из основного цикла делается 1 сравнение

Тогда:

* В лучшем случае:
* В худшем случае:

1. Отнесём к классу эффективности:

* Лучший случай: ***линейный класс***
* Худший случай: ***квадратичный класс***

IV. Эмпирический анализ алгоритма

Воспользуемся общим планом эмпирического анализа эффективности алгоритмов (*источник [4]*)*:*

1. Цель эксперимента: проверка точности теоретических выводов об эффективности алгоритма
2. Измеряемая метрика ***f***: трудоёмкость алгоритма. Единицы измерения: кол-во выполненных базовых операций

***\*****Базовая операция: сравнение*

***\****Способ измерения: *счётчик*

1. Диапазоны значений:
   1. Кол-во вершин ***n*** (*задаёт размер входных данных*):
   2. Кол-во рёбер:
   3. Веса рёбер:
   4. Номер вершины:
2. Программная реализация:

* Генератора образца входных данных:

*На вход принимает кол-во вершин* ***n****, на выходе даёт матрицу инцидентности и номер стартовой вершины.*

Основные шаги:

* + Случайным образом выбирается кол-во рёбер ***m*** (в диапазоне *b)*)
  + Случайным образом генерируется матрица инцидентности ***n*** x ***n***:
    1. Сначала задаётся полностью заполненная 0-ми весами (означает, что нет связей)
    2. Случайным образом выбирается ребро (два номера вершин (каждый в диапазоне *d*))
    3. Случайным образом выносится решение о наличие ребра (диапазон)
    4. Если *True*, то случайным образом выставляется вес (конечный поддиапазон диапазона *c)*)
  + Случайным образом назначается стартовая вершина (диапазон *d)*)
* Алгоритма:

*На вход принимает матрицу инцидентности и номер стартовой вершины, на выходе даёт массив кратчайших расстояний от стартовой вершины до всех остальных.*

Сам алгоритм описан во II части.

**\*** Ссылка на GitHub: <https://github.com/Dima12101/Empirical_Analysis>

V. Литература

1. *Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн*. Часть VI.Алгоритмы для работы с графами: Глава 24. Кратчайшие пути из одной вершины: Алгоритм Дейкстры //Алгоритмы: построение и анализ — 3-е изд. — М.: «Вильямс», 2013. — С. 696–702. — ISBN 978-5-8459-1794-2.
2. Kvodo (Computing Science & Discrete Match). Алгоритм Дейкстры [Электронный ресурс]. – Режим доступа: http://kvodo.ru/dijkstra-algorithm.html, свободный – (24.11.2019).
3. *Левитин А. В*. Глава 9. Жадные методы: Алгоритм Дейкстры // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 386–391. — ISBN 5-8459-0987-2.
4. *Левитин А. В*. Глава 2. Основы анализа эффективности алгоритмов: Математический анализ нерекурсивных алгоритмов // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 98–106. — ISBN 5-8459-0987-2.
5. *Левитин А. В*. Глава 2. Основы анализа эффективности алгоритмов: Эмпирический анализ алгоритмов // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 127–134. — ISBN 5-8459-0987-2.