САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

*Кафедра информатики и прикладной математики*

**Лабораторная работа №2**

*«Алгоритм Дейкстры на d-куче»*

Выполнил:

студент II курса группы 2125

Припадчев Артём

Проверил:

Зинчик А.А.

Санкт-Петербург

2014

**Задание:** написать программу реализующую алгоритм Дейкстры на основе 4- и 5-кучи. Написать программу, реализующую эти алгоритмы, для проведения экспериментов в которой можно выбирать:

* число n вершин и число m ребер графа,
* натуральные числа q и r, являющиеся соответственно нижней и верхней границей для весов графа.

Выходом данной программы должно быть время работы ТА алгоритма на основе 4-кучи и время работы TB алгоритма на основе 5-кучи.

Провести эксперимент на основе следующих данных: n=104+1, m=0,….,107 с шагом 105, q=1, r=106. Нарисовать полученные графики времени выполнения от количества ребер.

**Описание алгоритма**

Описание структур данных

В данной лабораторной работе рассматривается задача нахождения кратчайших путей в ориентированном графе без петель и кратных ребер. Соответственно основной структурой данных является граф, который представлен как массив списков смежности (окрестностей) своих вершин. При этом в окрестности хранятся структуры с номерами вершин, в которые идут ориентированные ребра. Каждая структура хранит номер вершины, оканчивающей ребро, вес идущего в нее ребра, признак принадлежности вершины графу и указатель на следующую вершину в окрестности. Окончанием окрестности является нулевая ссылка.

В алгоритмах используется также приоритетная очередь на основе Д-кучи. Д-куча представлена как три массива: массив имен вершин, массив ключей вершин (где ключ вершины – это оценка кратчайшего расстояния от исходной вершины до данной) и массив индексов вершин (то есть номеров вершин в приоритетной очереди). Приоритетная очередь на основе Д-кучи строится так: Д-куча представляется как корневое дерево, где корень находится на наивысшем уровне. Корнем является вершина с наименьшим ключом. Каждая вершина (не лист), кроме, возможно, одной имеют ровно Д потомков. Ключ каждого потомка вершины не меньше ключа самой вершины. Таким образом, высота Д-кучи равна [logdn]. С Д-кучей осуществляются следующие операции: всплытие вершины (вершина с ключом, меньшим, чем у родителя, поднимается наверх, пока не займет соответствующее ее ключу положение), погружение вершины (вершина с ключом, большим, чем у потомка, опускается вниз, пока не займет соответствующее ее ключу положение), изъятие минимума (выбор и удаление вершины с наименьшим ключом), образование приоритетной очереди. Для каждой вершины Д-кучи можно определить ее первого, последнего и минимального потомка и родителя.

В данной лабораторной работе рассматриваются два алгоритма: алгоритм Дейкстры, основанный на использовании 4-кучи и алгоритм Дейкстры, основанный на использовании 5-кучи. Оба алгоритма возвращают два массива значений: массив расстояний от исходной вершины до всех остальных и массив «отцов» (последних вершин на пути от исходной). Т.к. алгоритмы различаются только значением числа Д, а в остальном они одинаковы, рассмотрим общую схему их работы.

**Алгоритм Дейкстры, реализованный на основе Д-кучи**

В данном алгоритме сначала происходит создание Д-кучи, а затем инициализация массивов, хранящих Д-кучу и предварительная инициализация возвращаемых массивов. Массив «отцов» инициализируется нулями, массивы расстояний и ключей в Д-куче инициализируются неким «большим» значением (это значение должно быть больше длины любого пути в графе), массивы имен и индексов в Д-куче инициализируются последовательно натуральными числами от 1 до количества вершин в графе. Ключ исходной вершины устанавливается как ноль. Далее на основе Д-кучи организуется приоритетная очередь.

Пока приоритетная очередь не исчерпана, выполняются следующие действия: извлекается минимум, расстояние до данной вершины устанавливается равным ее ключу, далее выполняется просмотр всей окрестности этой вершины. В ходе просмотра окрестности, пока не достигнут конец окрестности, для каждой вершины выполняются следующие действия: сравнивается ключ текущей вершины с суммой расстояния от исходной вершины до вершины-родителя и веса ребра из вершины-родителя в текущую вершину. Если ключ больше, то он устанавливается равным данной сумме, затем происходит всплытие текущей вершины в приоритетной очереди, «отцом» текущей вершины устанавливается вершина-родитель.

Временная сложность алгоритма Дейкстры, реализованного на основе Д-кучи (Д является константой, не меньшей 2), оценивается сверху величиной O((d\*n+m)⋅logdn), где n - число вершин, m - число ребер.

**Результат работы**

В ходе лабораторной работы был проведен эксперимент, в котором изменялся один из параметров исходного графа с определенным шагом. Результат работы представлен на соответствующем графике. На оси X отложены значения изменяющегося параметра, а по оси Y время в миллисекундах.

В обозначениях серии используются следующие условные обозначения: n – число вершин графа, m – число ребер графа, r – верхняя граница веса ребра, q – нижняя граница веса ребра.

* n=104+1, m=0,….,107 с шагом 105, q=1, r=106

**Выводы:**

* Время работы алгоритма основанного на 5-куче практически совпадает с временем алгоритма на 4-куче. Едва ли заметно то, что время на 5-куче (линия B на графике) в большинстве случаев чуть меньше времени на 4-куче. Это объясняется тем, что количество вершин невелико, и на них значение 4\*n⋅log4n практически равно значению 5\*n⋅log5n.
* Зависимость от количества вершин получилась линейная, что соответствует теоретической временной сложности.