Алгоритм Дейкстры

Описание алгоритма

Алгоритм Дейкстры – это алгоритм на графах, решающий задачу нахождения самого кратчайших путей от одной из вершин графа до всех остальных. Работает только если в графе нет рёбер отрицательного веса.

Для каждой вершины графа создаётся переменная – метка. В начале работы алгоритма метка исходной вершины равна нулю, метки всех остальных вершин – бесконечности.

В процессе работы на каждой итерации алгоритм посещает одну из вершин. Среди вершин графа, которые не посещались ранее, выбирается вершина с минимальным значением метки (на первой итерации это всегда начальная вершина). Далее для каждой вершины, имеющей с посещаемой на данной итерации общее ребро и не посещённой ранее, вес общего ребра складывается с весом метки посещаемой вершины. Если вычисленная сумма меньше собственной метки смежной вершины, оно записывается в эту метку.

Итерации повторяются до тех пор, пока все вершины графа не будут посещены.

Реализация алгоритма

В ходе работы алгоритм реализован на языке C++.

Для этого заведены следующие классы:

1. Класс List – хранит в себе список вершин графа, соединённые с вершиной, к которой относится экземпляр класса, и веса соответствующих рёбер (так как связи между парой вершин хранятся в независимых экземплярах класса List, рёбра могут иметь разный вес в разных направлениях или вообще быть односторонними).
2. Класс Point – экземпляр данного класса соответствует одной вершине графа. Содержит метку, номер вершины при движении из которой эта метка получена, а также экземпляр класса List со списком выходящих из вершины рёбер. Данный класс содержит в себе все необходимые методы для исполнения алгоритма Дейкстры и вывода результатов.

Полный код классов с комментариями, а также пример использования представлены в приложении 1.

Анализ алгоритма

Пусть n – число вершин графа, а m – число рёбер. В ходе работы алгоритм последовательно проходит все вершины по одному разу, то есть основной цикл выполняется n раз. На каждой итерации алгоритм проверяет, все ли вершины посещены, после чего ищет непосещённую вершину с минимальной меткой, в обоих случаях проходя последовательно все вершины. Далее для найденной вершины с минимальной меткой алгоритм проходит каждое ребро. Так как ребра соединены с двумя вершинами, каждое ребро проходится дважды, поэтому всего алгоритм производит проход по ребру 2m раз. Таким образом, время работы алгоритма равно k1∙n2 + k2∙m, а значит после пренебрежения константами временная сложность – O(n2 + m).