Алгоритм обхода графа в глубину:

Данный алгоритм можно сравнить с разгорающимся лесом (это отразилось на названии переменных в коде, см. ниже). Зажжем первую вершину (выбираем какую-то стартовую). От нее, как по мостикам, по ребрам графа огонь (наш алгоритм) переходит на соседние вершины. После этого сами соседние вершины уже зажигают своих соседей. При этом,чтобы наш алгоритм постоянно не записывал уже зажженные вершины, нам нужно,чтобы вершины «догорали» (необходимо убирать вершины, соседей которых уже проверили, из списка зажженных вершин). Те этот алгоритм вкратце можно описать так:

1) Вершина графа загорается.

2) Передает огонь на все своих соседей

3) Догорает

И так происходит до того момента,пока мы не пройдем по всем соседям.

Перейдем к реализации программы:

2) Создаем множество «догоревших» вершин,если мы еще этого не сделали.

4) Добавляем стартовую вершину в «догоревшие».

5) Хранение времени,за которое мы доходим до вершины (это позволяет находить расстояние до вершины от стартовой).

6) Добавление вершины в очередь «горящих» (по нему мы идем).

7) Цикл работает,пока есть хотя одна горящая вершина.

8) Берем первую вершины из очереди

9) Проходим по всем соседям вершины

11) Добавляем нашу вершину в очередь «зажженных»

12) Добавляем в «догоревших» (чтобы снова не записывать его в очередь «зажженных»

13) Это позволяет получить остовное дерево.

14) Так и получаем расстояние до вершины.

Алгоритм Дейкстры:

Данный алгоритм необходим для поиска кратчайшего маршрута от исходной вершины до всех вершин графа. Его метод работы основан на обходе графа в ширину (как мы помним, данный алгоритм позволяет находить расстояния в навзвешенном графе). Идея аналогичная - «зажигаем» вершины, после этого проходим по соседям «зажженной» вершины. Но теперь в нашем проходе появляется определенный порядок - в начале мы идем к вершинам, путь к которым наиболее краток. Это позволяет раньше добавить вершины в список «сгоревших» и закончить вычислять длину маршрута к ним.

Перейдем к реализации алгоритма:

1) G - словарь словарей (каждой вершине соответствует список вершин, каждой из которых поставили в соответствие длину ребра).

2) Задаем изначальный словарь длин расстояний до вершин.

3) Задаем длину расстояния до исходной вершины (она ноль, очевидно).

4) Задаем множество пройденных вершин.

5) Пока не все вершины использованы.

6) Задаем расстояние (делаем его бесконечным, чтобы потом можно было найти минимальное).

7-10) Находим вершину с минимальным путем до нее. При первом проходе это будет начальная вершина (до нее путь ноль). При втором проходе она уже будет смотреть и выбирать из соседей исходной вершины и тд.

11-14) Пробегаемся по соседям нашей вершины и рассчитываем путь до нее. Если он меньше того пути, который сейчас соответствует этой вершине, то мы записываем новое значение - это путь до той вершины, по соседям которой мы пробегаемся плюс путь от нее до соседа.

15) Добавляем проверенную вершину в использованное.

Программа 4.2 имеет такую же схему работы, но там для большей эффективности использована пирамида кортежей.