

Формулировка задачи. Дан непустой граф. Необходимо найти путь между двумя вершинами содержащий наименьшее количество вершин (ребер).

Идея алгоритма

Перед началом работы алгоритма каждой вершине присваивается число называемое волновой меткой и имеющее минимальное значение (в алгоритме это минимальное значение равно -1). Работа алгоритма заключается в распространении фронта волны от исходной вершины по всему графу, либо до полного затухания волны либо до достижения вершины назначения. В процессе движения волны значения волновых меток изменяются по определенному правилу позволяющему определять длину пути до каждой вершины через которую прошла волна. Правило заключается в увеличении значения волновой метки вершины в момент прохождения волны через неё. При этом значение приращения волновых меток зависит от длины пути пройденного волной. Таким образом можно утверждать, что чем больше значение волновой метки, тем больший путь был пройден волной до данной вершины.

Для полного понимания идеи внимательно прочитайте примечание данное после текста алгоритма.

Исходные структуры данных – множество вершин каждой из которых присвоена волновая метка с начальным значением -1.

Текст алгоритма

- метка исходной точки = 0
- Смещение = 0
- фронт волны состоит только из исходной точки
- новый фронт пуст
- Повторять
 - Для всех ВЕРШИН фронта
 - Для всех вершин инцидентных ВЕРШИНЕ
 - Если метка = -1 ТО метка = Смещение + 1 и вершина включается в список вершин нового фронта
 - фронт = новый фронт
 - Смещение = Смещение + 1
- Если Новый фронт пуст или достигнута вершина назначения завершить выполнение цикла
- Если Новый фронт пуст то решения нет иначе решение найдено.

Примечание. Важно заметить, что значение метки меняется только в том случае если волна в неё пришла первый раз. Это даёт возможность сохранять пути к вершине назначения в чистом виде. Но конечно эта особенность делает механизм распространения волны совершенно не похожим на распространение физической волны.

После завершения работы алгоритма искомый путь восстанавливается следующим алгоритмом:

- Очередная вершина есть вершина назначения.
- Пока текущая вершина не есть исходная делать
 - Новой Очередной вершиной становится вершина чьё значение метки на единицу меньше значения метки Очередной вершины.

