

## Волновой алгоритм.

Поиск кратчайшего пути между вершинами  $V_1$  и  $V_2$  в невзвешенном графе выполняется при помощи алгоритма поиска в ширину. Этот алгоритм имеет также название «волнового алгоритма», который впервые был открыт в 1961 г. и широко применяется в разработке печатных плат для электронных устройств.

Алгоритм состоит из двух этапов:

- «распространения волны»
- восстановление обратного пути

Первый этап заключается в том, что от исходной вершины  $V_1$  мы начинаем «двигаться» по принципу распространения круговой волны: сначала по смежным с  $V_1$ . Под «движением» здесь понимается просмотр достижимых (т. е. смежных) вершин и присваивание им метки «фронта волны». Все смежные с  $V_1$  вершины будут помечены единицей. Далее, найдем множество  $\{M\}$  всех непомеченных вершин, смежных со всеми вершинами прежнего «фронта». Через эти вершины пройдет «фронт» с пометкой 2. Таким образом, увеличивая на каждом шаге метку фронта волны, мы будем двигаться по графу, удаляясь от исходной вершины  $V_1$  и в конечном итоге, на некотором шаге множество  $\{M\}$  окажется пустым.

Если при этом метка для вершины  $V_2$  не будет установлена, это означает, что путь между вершинами  $V_1$  и  $V_2$  отсутствует (т. е. граф не связный).

Как видим, первый этап представляет из себя решение предыдущей задачи (вычисление «уровня потомства» вершин относительно заданной).

На втором этапе (в том случае, если для вершины  $V_2$  успешно установлена метка) мы «восстанавливаем путь» из последней вершины в первую. Обратный ход используется поскольку справедливо следующее утверждение:

«Кратчайший путь из любой вершины  $V_i$  (с меткой расстояния  $L_{Vi}$ ) до вершины  $V_1$  лежит через любую из вершин с меткой  $L_{Vi}-1$ , смежную с  $V_i$ ».

Квантор общности в этом утверждении позволяет нам построить циклический алгоритм, переходя от вершины к вершине по кратчайшему маршруту, пока не достигнем вершины  $V_1$ . При неформальном рассмотрении второй этап алгоритма выглядит следующим образом:

1. Создадим пустой маршрут. Если метка вершины  $V_2$  не установлена, то **завершим алгоритм.**
2. Сделаем вершину  $V_2$  текущей:  $V_{CUR}=V_2$ . Сделаем уровень  $L_{V_2}$  текущим:  $L_{CUR}=L_{V_2}$
3. Добавим к вершине  $V_{CUR}$  текущий маршрут
4. Если  $L_{CUR}=1$ , то добавим к вершине  $V_1$  текущий маршрут и **завершим алгоритм.**
5. Получим множество  $\{S\}$  вершин, смежных с  $V_{CUR}$ .
6. Выберем из множества  $\{S\}$  любую вершину  $V_{NEXT}$ , имеющую метку  $L_{CUR}-1$ .  
(*Замечание: гарантированно существует одна или более таких вершин.*)
7. Сделаем вершину  $V_{NEXT}$  текущей:  $V_{CUR}=V_{NEXT}$ .
8. Уменьшим уровень  $L_{CUR}$  на единицу
9. Перейдем к шагу 2.