

## **Поиск в глубину**

При *поиске в глубину* посещается первая *вершина*, затем необходимо идти вдоль ребер графа, до попадания в *тупик*. *Вершина* графа является *тупиком*, если все смежные с ней вершины уже посещены. После попадания в *тупик* нужно возвращаться назад вдоль пройденного пути, пока не будет обнаружена *вершина*, у которой есть еще не посещенная *вершина*, а затем необходимо двигаться в этом новом направлении. Процесс оказывается завершенным при возвращении в начальную вершину, причем все смежные с ней вершины уже должны быть посещены.

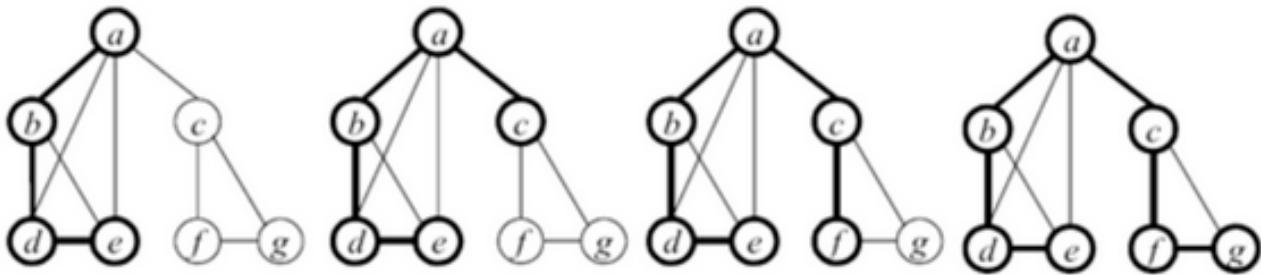
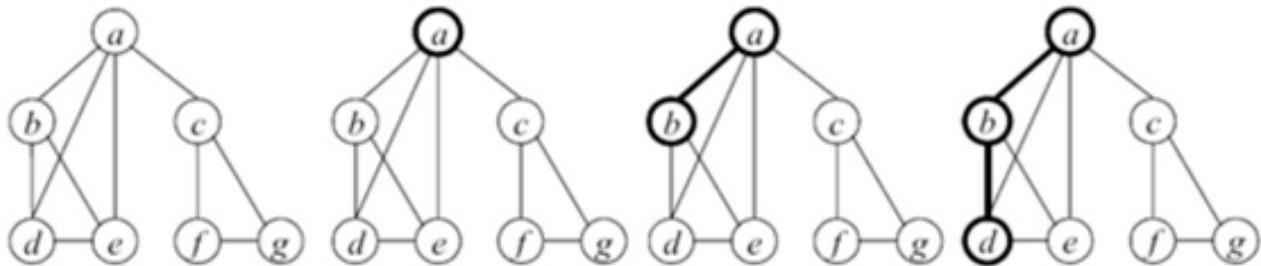
Таким образом, основная идея поиска в глубину – когда возможные пути по *ребрам*, выходящим из вершин, разветвляются, нужно сначала полностью исследовать одну ветку и только потом переходить к другим веткам (если они окажутся нерассмотренными).

### ***Алгоритм поиска в глубину***

Шаг 1. Всем вершинам графа присваивается *значение* не посещенная. Выбирается первая *вершина* и помечается как посещенная.

Шаг 2. Для последней помеченной как посещенная вершины выбирается смежная *вершина*, являющаяся первой помеченной как не посещенная, и ей присваивается *значение* посещенная. Если таких вершин нет, то берется предыдущая помеченная *вершина*.

Шаг 3. Повторить шаг 2 до тех пор, пока все вершины не будут помечены как посещенные



## **Поиск в ширину**

При *поиске в ширину*, после посещения первой вершины, посещаются все соседние с ней вершины. Потом посещаются все вершины, находящиеся на расстоянии двух ребер от начальной. При каждом новом шаге посещаются вершины, *расстояние* от которых до начальной на единицу больше предыдущего. Чтобы предотвратить повторное посещение вершин, необходимо вести *список* посещенных вершин. Для хранения временных данных, необходимых для работы алгоритма, используется *очередь* – упорядоченная последовательность элементов, в которой новые элементы добавляются в конец, а старые удаляются из начала.

Таким образом, основная идея *поиска в ширину* заключается в том, что сначала исследуются все вершины, смежные с начальной вершиной (*вершина* с которой начинается обход). Эти вершины находятся на расстоянии 1 от начальной. Затем исследуются все вершины на расстоянии 2 от начальной, затем все на расстоянии 3 и т.д. Обратим внимание, что при этом для каждой вершины сразу находятся *длина* кратчайшего маршрута от начальной вершины.

### **Алгоритм поиска в ширину**

Шаг 1. Всем вершинам графа присваивается *значение* не посещенная. Выбирается первая *вершина* и помечается как посещенная (и заносится в *очередь*).

Шаг 2. Посещается первая *вершина* из очереди (если она не помечена как посещенная). Все ее соседние вершины заносятся в *очередь*. После этого она удаляется из очереди.

Шаг 3. Повторяется шаг 2 до тех пор, пока *очередь* не пуста

