**1**

Вступление

**2**

Отгадайте, о каких графах мы будем говорить

**3**

Наш план

**4**

Давайте посмотрим определения (граф, обход графа) Артурчик, сфоткай.

**5**

Существует 2 вида обхода графа: в глубину (Depth-First Search) и в ширину (Breath-First Search). Начнем с обхода в глубину.

Рассказать про обход в глубину. Артурчик, сфоткай.

**6-12**

Предположим, что у нас есть ориентированный граф, который выглядит так.

Мы находимся в точке «s» и нам нужно найти вершину «t». Применяя DFS, мы исследуем один из возможных путей, двигаемся по нему до конца и, если не обнаружили t, возвращаемся и исследуем другой путь. Вот как выглядит процесс:

**7**

Здесь мы двигаемся по этому пути к ближайшей вершине и видим, что это не конец пути. Поэтому мы переходим к следующей вершине.

**8**

Мы достигли конца пути, но не нашли t, поэтому возвращаемся в s и двигаемся по второму пути.

**9**

Достигнув ближайшей к точке «s» вершины пути 2 мы видим три возможных направления для дальнейшего движения. Поскольку одну вершину вершину мы уже посещали, то двигаемся по второму.

**10**

Мы вновь достигли конца пути, но не нашли t, поэтому возвращаемся назад и следуем по третьему пути.

**11**

наконец, достигаем искомой вершины «t». (переключить на слайд **12)**

**13**

Так работает DFS. Двигаемся по определенному пути до конца. Если конец пути — это искомая вершина, мы закончили. Если нет, возвращаемся назад и двигаемся по другому пути до тех пор, пока не исследуем все варианты.

Мы следуем этому алгоритму применительно к каждой посещенной вершине.

Необходимость многократного повторения процедуры указывает на необходимость использования рекурсии для реализации алгоритма.

Также вы можете увидеть реализацию обхода в глубину на C++.Артурчик, сфоткай.

**14**

Рассказать про обход в ширину. Артурчик, сфоткай.

**15**

BFS следует концепции «расширяйся, поднимаясь на высоту птичьего полета» («go wide, bird’s eye-view»). Вместо того, чтобы двигаться по определенному пути до конца, BFS предполагает движение вперед по одному соседу за раз. Это означает следующее…

У нас снова есть ориентированный граф. Давайте увидим как работает обход в ширину.

Вместо следования по пути, BFS подразумевает посещение ближайших к s соседей за одно действие (шаг), затем посещение соседей и так до тех пор, пока не будет обнаружено t.

(Переключать медленно слайды **16, 17, 18**)

**19**

Далее возникает вопрос: как узнать, каких соседей следует посетить первыми?

Для этого мы можем воспользоваться концепцией «первым вошел, первым вышел» (first-in-first-out, FIFO) из очереди (queue). Мы помещаем в очередь сначала ближайшую к нам вершину, затем ее непосещенных соседей, и продолжаем этот процесс, пока очередь не опустеет или пока мы не найдем искомую вершину. Артурчик, сфоткай.

**20**

На экране время выполнения…

Может показаться, что BFS работает медленнее. Однако если внимательно присмотреться к визуализациям, можно увидеть, что они имеют одинаковое время выполнения.

Очередь предполагает обработку каждой вершины перед достижением пункта назначения. Это означает, что, в худшем случае, BFS исследует все вершины и грани.

Несмотря на то, что BFS может казаться медленнее, на самом деле он быстрее, поскольку при работе с большими графами обнаруживается, что DFS тратит много времени на следование по путям, которые в конечном счете оказываются ложными. BFS часто используется для нахождения кратчайшего пути между двумя вершинами.

Таким образом, время выполнения BFS также составляет O(V + E), а поскольку мы используем очередь, вмещающую все вершины, его пространственная сложность составляет O(V). Артурчик, сфоткай.

**21**

Теперь поговорим про связанность в ориентированных и неориентированных графах. Определение и пример на экране. Артурчик, сфоткай.

**22**

Это скучное, официальное определение орграфа и неорграфа. Артурчик, сфоткай.

**23**

А теперь давайте расскажем вам понятными словами. Зачитать определение. Пример на экране.

**24**

Теперь давайте разберемся со связанностью в орграфе и неорграфе (зачитать определения). Артурчик, сфоткай.

**25**

Теперь перейдем к компоненте связанности. (зачитать определение). Пример на экране. У неориентированного графа, изображенного на этом рисунке две компоненты связности. Первая компонента связности включает вершины 1, 2, 4, 5, а вторая состоит из одной вершины (3). Артурчик, сфоткай.

**26**

Пример на экране. У ориентированного графа, изображенного на этом рисунке две компоненты сильной связности. Первая компонента связности включает вершины 1, 2, 3, 5, а вторая состоит из одной вершины (4).

Для любой пары вершин из первого компонента существует хотя бы один путь, соединяющий эти вершины. Из вершины 4 нет пути ни в одну вершину графа. Артурчик, сфоткай.