

dz prog

Маслов Никита

Февраль 2025

## 1 Ответы на вопросы

### 1. Что такое граф?

Граф - это совокупность ребер и вершин, которые необязательно соединены друг с другом, то есть у нас может быть условно 3 точки, которые будут являться графом.

### 2. В случае простого графа (отсутствуют петли и кратные ребра), какое максимальное возможное количество ребер $M$ ? Зависит ли это число от ориентированности графа?

Ориентированный -  $M(M - 1)$ , неориентированный -  $\frac{M(M-1)}{2}$ , зависит, так как в ориентированном графе  $AB$  и  $BA$  - два разных ребра  $\Rightarrow$  в ориентированном графе ребер больше.

### 3. Как по заданной матрице смежности быстро проверить граф на ориентированность?

Если матрица не меняется при транспонировании, то граф неориентированный, в ориентированном не всегда так.

### 4. Как изменятся структуры списка ребер и списка смежности в случае взвешенного графа?

В список ребер добавляется вес третьим компонентом, в списке смежности вместе с номером вершины будет идти вес.

### 5. Что такое компонента связности графа? Каким может быть максимальное и минимальное кол-во компонент в одном графе?

Компонента связности - связанный "подграф" графа, то есть в графе может быть несколько компонент, то есть несколько не связанных друг с другом

подграфов. Максимальное кол-во -  $n$ , где  $n$  - кол-во вершин, минимальное - 1, сам граф.

## **6. Можно ли с помощью BFS искать циклы в графе? Выгоднее ли это чем поиск через DFS?**

Нет, так как с помощью BFS мы не возвращаемся в пройденную вершину, а обычно идем вниз по уровням, поэтому мы не имеем представления, вернулись ли мы в вершину, в которой были, или же нет, поэтому нельзя найти цикл, в отличие от DFS, в котором мы можем обозначить непройденные вершины белым цветом, а посещенные - серым. В DFS мы обходим граф вниз и при достижении тупика, обозначаем его черным цветом, а если у нас получилось так, что мы вернулись в серую вершину, то обозначаем ее черным, тогда путь, пройденный от нее до нее же, будет циклом. Поэтому лучше искать циклы с помощью DFS.

## **7. Зачем вводится требование на неотрицательность веса ребра в алгоритме Дейкстры? Можно ли заставить алгоритм корректно работать в случае отрицательных весов (без их изменения).**

Неотрицательность веса ребра вводится для того, чтобы мы могли ввести минимальное расстояние до точки, которое будет одним ребром, иначе у нас может быть обходной путь через отрицательные ребра, путь которых в совокупности может быть меньше ребра до вершины. Можно добавить каждому ребру большое число, чтобы все стали положительными, а потом вычесть это на кол-во ребер, пройденных за путь.

## **8. Зачем запускать алгоритм Форда-Беллмана в N-й раз?**

Если на N-й раз вес изменится, то есть отрицательный цикл.

## **9. Подумайте, в каком случае Форд-Беллман может работать быстрее чем Дейкстра. Каким образом можно оптимизировать эту ситуацию?**

Форд-Беллман может работать быстрее чем Дейкстра, когда есть отрицательные ребра, если у графа маленькое количество вершин, то, возможно, быстрее Дейкстры. Если при обходе графа вес не изменился, то можно заканчивать алгоритм.

10. В каких ситуациях вы предпочтете рекурсивную реализацию DFS итеративной и наоборот? Эффективно ли применение списка в качестве стека в итеративной версии? Что насчет очереди в случае BFS?

ПОТОМ