

Контрольные вопросы

1. Граф — это структура, состоящая из вершин и рёбер, которые соединяют пары вершин.
2. В простом неориентированном графе с N вершинами максимум $M = \frac{N(N-1)}{2}$ рёбер. Для каждой из N вершин можно направить ребра ко всем оставшимся $N-1$ вершинам. При пересчете количества ребер для каждой вершины получаем удвоенное их количество, так как каждое ребро считаем дважды для каждой вершины. В ориентированном графе это число равно $M = N \cdot (N - 1)$, так как каждое ребро направлено от одной вершины к другой.
3. Чтобы проверить граф на ориентированность по матрице смежности, нужно сравнить её с транспонированной. Если $A^T = A$, то есть $A[i][j] = A[j][i]$ для всех i и j , то граф неориентированный. В случае, если матрица не симметричная, то граф является ориентированным.
4. В списке ребер, помимо двух соединенных друг с другом ребром вершин, будет также храниться и вес ребра, соединяющий вершины. В списке смежности для каждой вершины будет храниться соседняя вершина и вес ребра, соединяющая эти две соседние вершины.
5. Компонентой связности называется максимальное подмножество вершин, между которыми существует путь, то есть все вершины в компоненте связности друг от друга. Минимальное количество компонент связности в графе равно 1, что соответствует связному графу. В таком графе существует путь между любой парой вершин. Максимальное количество компонент в графе равно количеству вершин графа, то есть каждая вершина является изолированной.
6. BFS может обнаружить цикл, если при обходе вершины мы попадаем в уже посещённую вершину, которая не является родителем текущей. В DFS проще отслеживать наличие циклов через статус вершин, то есть была ли вершина обработана и посещена. DFS строит рекурсивный стек, так что проще отследить предыдущие вершины. BFS в то же время может быть сложнее в реализации.
7. Изначально в алгоритме Дейкстры расстояния до всех вершин, кроме стартовой, не известны. До стартовой расстояние 0. На каждой итерации алгоритм среди непомеченных вершин выбирает вершину с наименьшим до нее

расстоянием, пометает вершину как посещенную и пытается улучшить расстояние до смежных с ней вершин. На каждой итерации поддерживается инвариант, что расстояния до помеченных вершин уже являются кратчайшими и более меняться не будут, поэтому граф не должен содержать ребра с отрицательным весом. При отрицательных весах оптимальность пути может быть нарушена, так что заставить алгоритм работать в случае отрицательных весов без их изменения не получится.

8. За первые $N-1$ проходов по всем ребрам алгоритм Форда-Беллмана гарантирует, что все кратчайшие пути найдены. N -й проход вместе с релаксацией помогает проверить на наличие отрицательного цикла: если есть улучшения, то в графе есть отрицательный цикл. Если цикл будет с отрицательной суммой весов, то по нему по нему можно будет бесконечно ходить и все время уменьшать длину, что что кратчайший путь будет не определен.
9. Форд-Беллман может работать быстрее на графах с маленьким количеством рёбер по сравнению с вершинами, то есть при разреженных графах или при наличии отрицательных весов, при которых алгоритм Дейкстры не осуществим. Оптимизировать можно, используя структуры данных, такие как двоичные кучи. При очень разреженных графах $M = N - 1$ разница между временем выполнения может быть небольшой. Если граф представляет из себя длинную цепочку, то есть линейный граф, то можем считать, что алгоритм Форда-Беллмана может пройти быстрее алгоритма Дейкстры. Алгоритм Дейкстры берем без двоичной кучи, так как тогда ее асимптотика будет в любом случае лучше, чем у Форда-Беллмана.
10. Рекурсивная версия DFS не может быть применена для больших графов, где в любом случае будет возникать переполнения стека рекурсии. Если же переполнения стека не происходит, то я бы использовал именно рекурсивную версию, так как она проще пишется, чем итеративная. При этом у итеративной реализации нет недостатков, свойственных рекурсивной версии. Так как удаление последнего элемента из стека происходит за $O(1)$, то можем сказать, что применения списка в качестве стека в итеративной версии вполне оправдано. При использовании списка в качестве очереди в итеративной версии BFS приведет к тому, что асимптотика удаления первого элемента возрастет до $O(N)$, что считается невыгодным.