

Что такое граф?

Граф — это просто набор вершин, которые соединены рёбрами. Он может быть ориентированным (со стрелочками) или неориентированным.

Максимальное количество рёбер в простом графе и его зависимость от ориентированности

Если в графе нет петель и кратных рёбер, то максимум рёбер в неориентированном графе — $\frac{N(N-1)}{2}$. В ориентированном — $N(N-1)$, так как каждое ребро может идти в обе стороны отдельно.

Проверка графа на ориентированность по матрице смежности

Нужно просто посмотреть на матрицу: если она симметрична (элементы a_{ij} и a_{ji} совпадают), то граф неориентированный. Если где-то есть несоответствие, значит, ориентированный.

Изменение списка рёбер и списка смежности во взвешенном графе

В списке рёбер теперь у каждого ребра будет вес (u, v, w) . В списке смежности у каждой вершины вместо списка соседей будет список пар (сосед, вес).

Компонента связности графа

Это кусок графа, где можно пройти от любой вершины до любой другой. Минимальное количество — 1 (если граф весь связный), а максимальное — N (если рёбер вообще нет).

Поиск циклов с помощью BFS и его эффективность

BFS может находить циклы, но удобнее использовать DFS, потому что он быстрее наткнётся на обратное ребро. BFS менее удобен, так как требует запоминания уровней вершин.

Неотрицательные веса в алгоритме Дейкстры

Если веса рёбер отрицательные, Дейкстра может выдавать неправильные результаты, так как всегда берёт минимальный путь. Если очень хочется работать с отрицательными весами, лучше использовать Форда-Беллмана.

Дополнительный запуск алгоритма Форда-Беллмана

Если прогнать алгоритм N раз, можно обнаружить отрицательный цикл. Если он есть, то кратчайший путь вообще нельзя корректно найти.

Когда Форд-Беллман работает быстрее Дейкстры

Если рёбер в графе мало, а вершины соединены длинными путями, Форд-Беллман может оказаться быстрее. Можно ускорить его с помощью очереди (SPFA-алгоритм).

Рекурсивный и итеративный DFS, использование структур данных

В Python стек рекурсии довольно большой, так что переполнения обычно не происходит. Тем не менее, итеративный DFS удобнее, если нужно явно управлять порядком обхода. В BFS всегда лучше использовать очередь, так как она естественным образом обходит граф по уровням.