Контрольное домашнее задание № 3

Кобызев Илья

Содержание

Введение

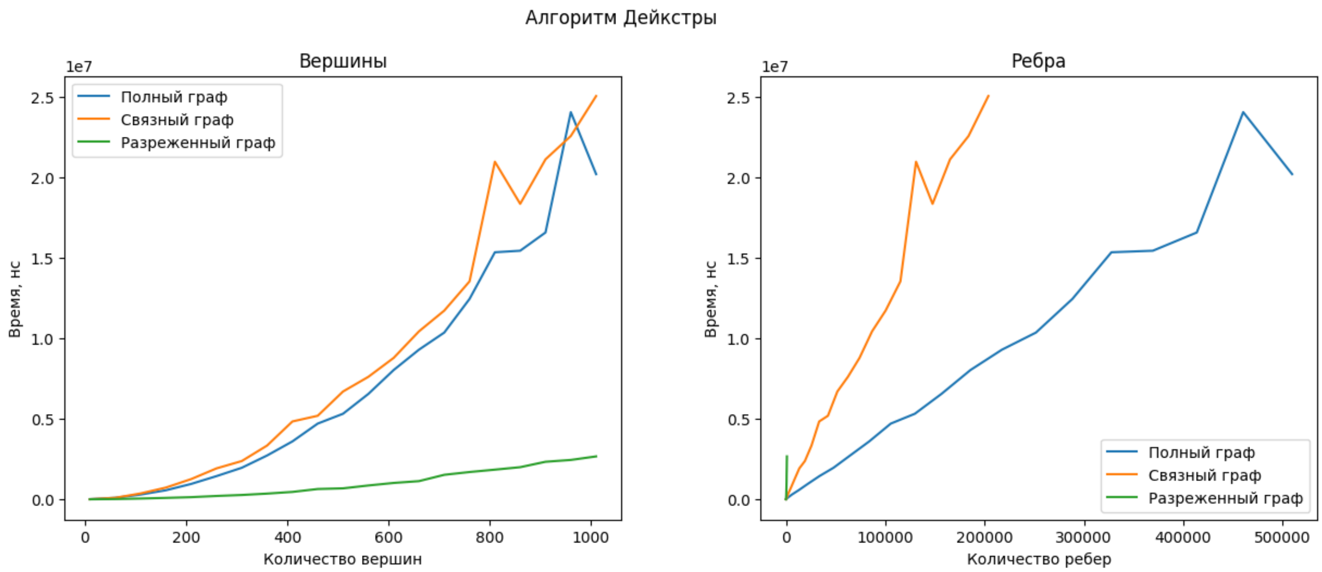
В отчете приведено время работы четырёх алгоритмов поиска кратчайшего пути в неориентированном взвешенном графе. Для измерения времени использовались 3 типа графов - полный, связный (коэффициент плотности 0.4) и разреженный. Результаты измерения времени поиска усреднялись 10 раз.

Алгоритмы поиска

1.1 Алгоритм Дейкстры

Алгоритм Дейкстры — это алгоритм для нахождения кратчайшего пути во взвешенном графе, который начинает с заданной вершины и постепенно обновляет расстояния до всех остальных вершин на основе их весов, выбирая каждый раз вершину с наименьшим расстоянием.

Асимптотическая сложность: $O(V^2)$.



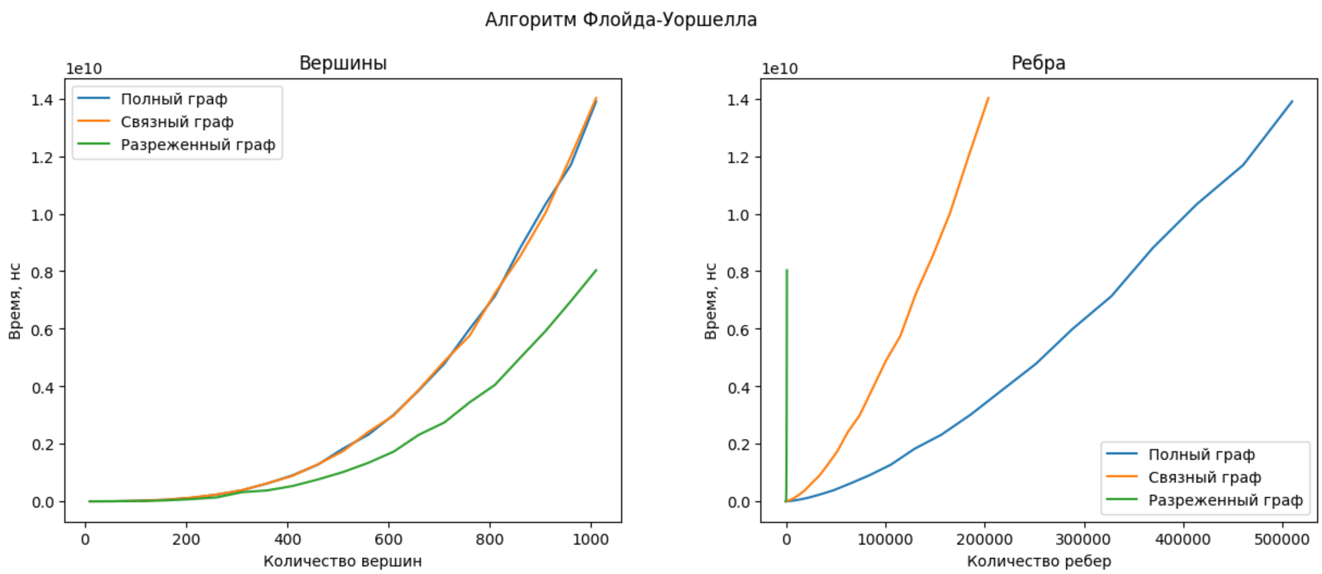
Как видно из графика, быстрее всего Алгоритм Дейкстры работает на разреженном графе, что объясняется гораздо меньшим количеством ребер.

Может показаться, что алгоритм работает дольше на связном графе, но это эффект масшаба – у связного графа меньше ребер. Из-за этого же кажется что график разреженного графа – вертикаль.

1.2 Алгоритм Флойда-Уоршелла

Алгоритм Флойда-Уоршелла — это алгоритм для нахождения кратчайших путей между всеми парами вершин во взвешенном ориентированном или неориентированном графе, который использует динамическое программирование для обновления расстояний через промежуточные вершины.

Асимптотическая сложность: $O(V^3)$.

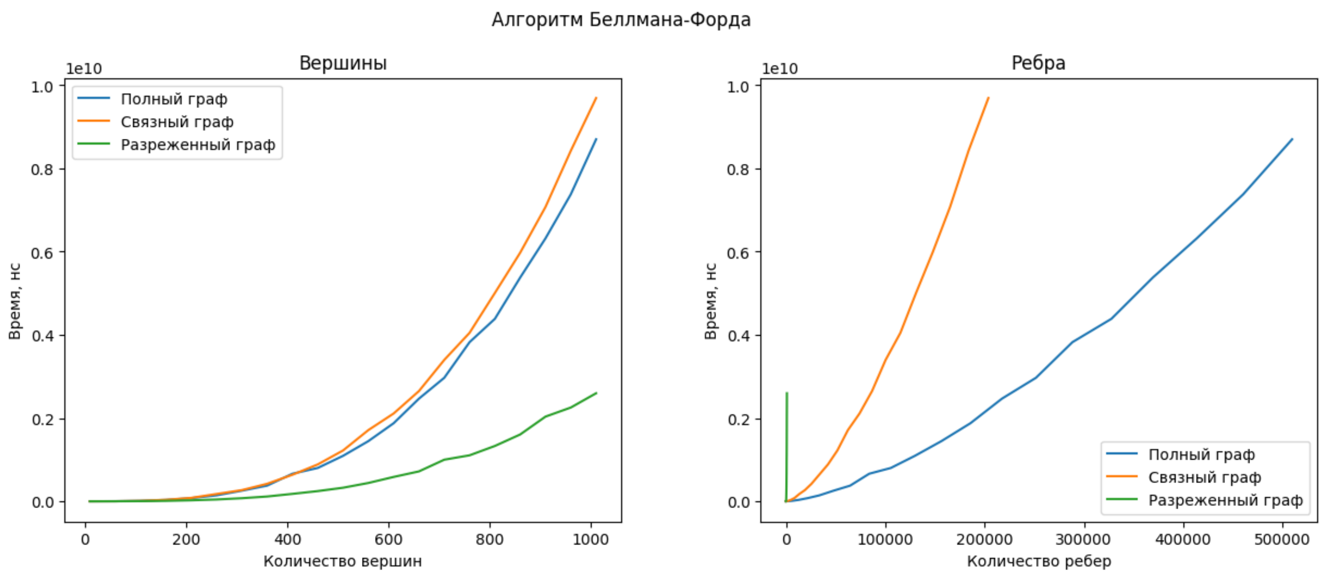


Теоретическая асимптотика подтвердилась экспериментально – на графиках явно виден куб. Немного опущенный график разреженного графа слева объясняется меньшим числом элементарных операций для сравнения.

1.3 Алгоритм Беллмана-Форда

Алгоритм Беллмана-Форда — это алгоритм для нахождения кратчайшего пути от одной вершины до всех остальных вершин во взвешенном графе. Он выполняет релаксацию ребер графа поочередно для всех вершин и повторяет этот процесс $V-1$ раз, где V – количество вершин, чтобы гарантировать нахождение кратчайших путей.

Асимптотическая сложность: $O(V \times E)$.

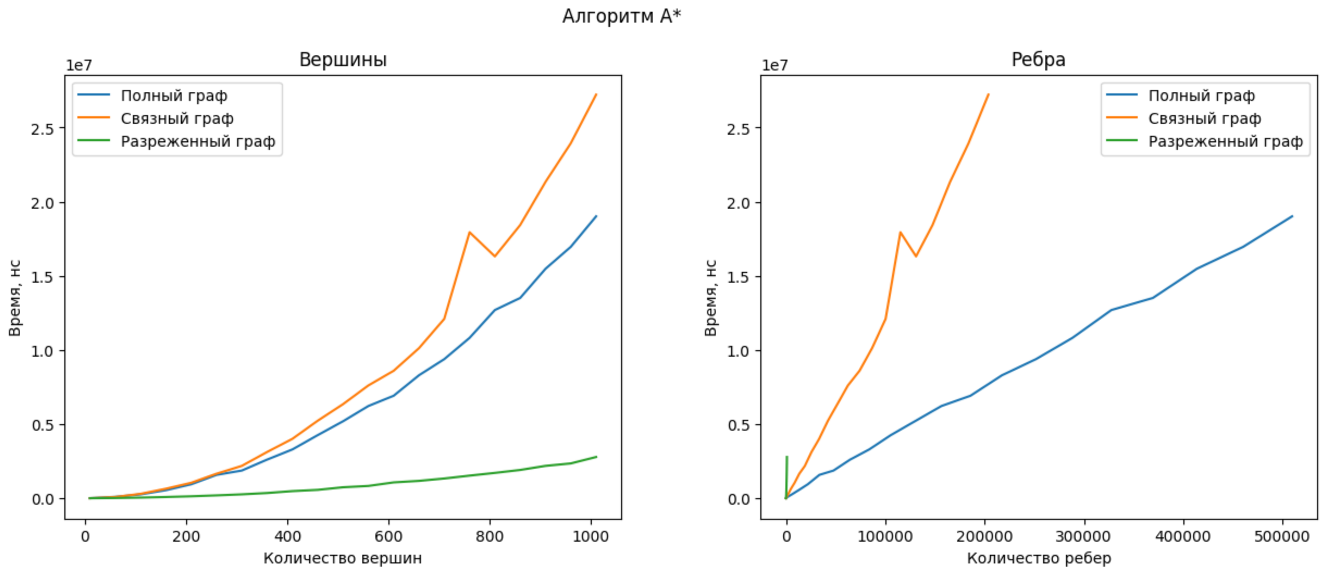


Теоретическая асимптотика подтвердилась экспериментально – разреженный граф с меньшим числом ребер имеет лучшее время.

1.4 Алгоритм A*

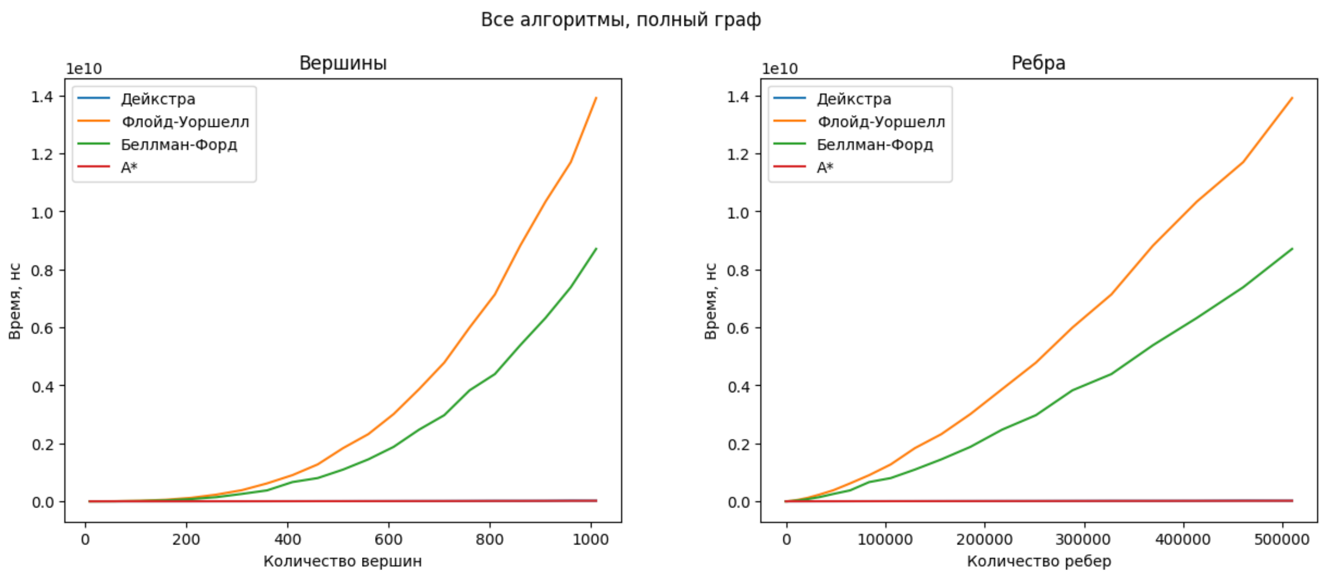
Алгоритм A* — это алгоритм поиска пути в графе, который комбинирует эвристику и информацию о стоимости перемещения, чтобы выбирать наилучший следующий шаг на основе оценки затрат до цели и уже пройденного пути.

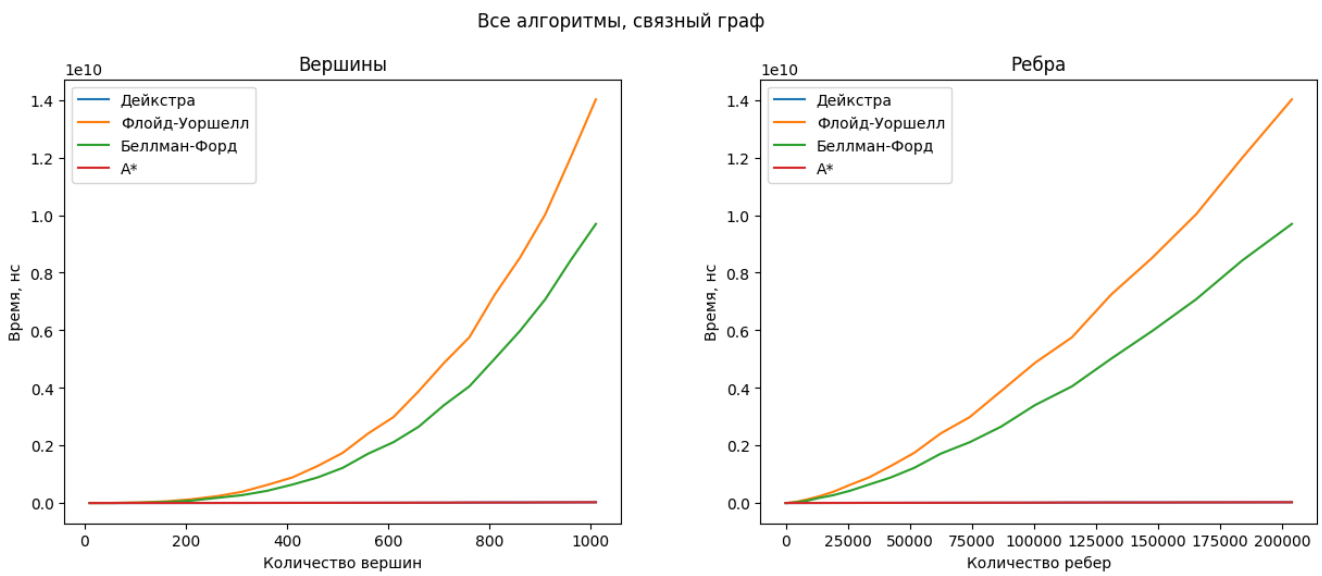
Асимптотическая сложность: $O((V + E) \times \log V)$.

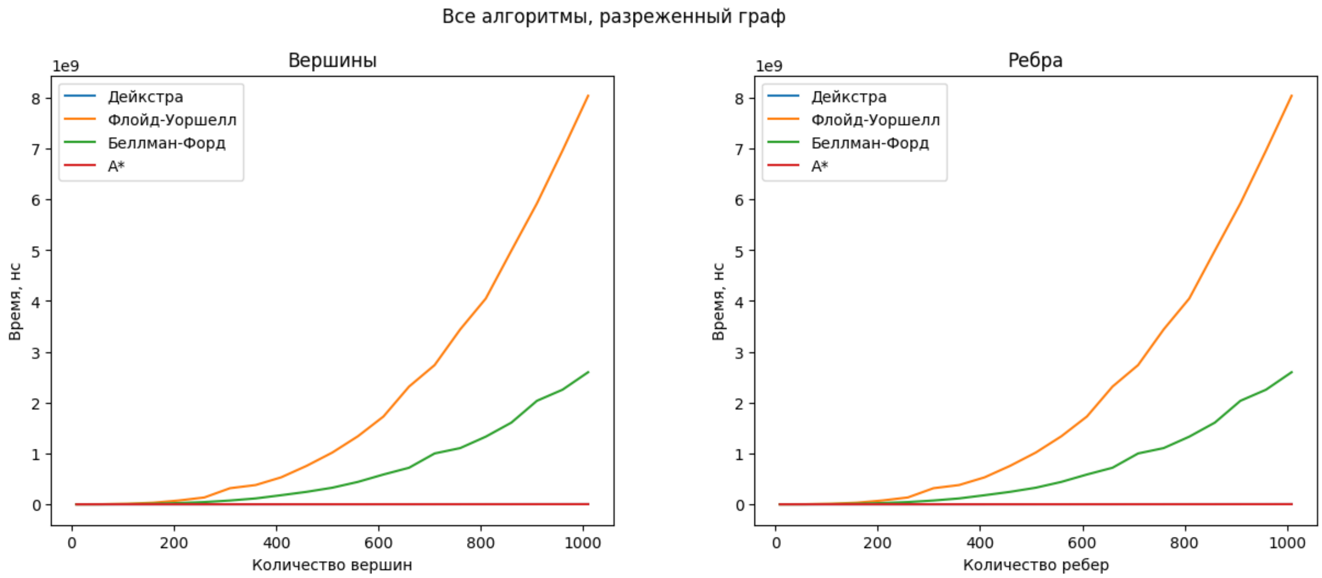


Судя по графику алгоритм работает быстрее на связном графе, чем на полном. Это может быть из-за эвристической функции: в плотном графе вероятнее всего будет более точное представление расстояний между вершинами, что позволяет эвристической функции более эффективно направлять поиск в нужном направлении и уменьшать количество проверок, или из-за приоритетной очереди: ребер больше, а значит вероятность нахождения нужной вершины ближе к начальной вершине повышается. Это может привести к меньшему количеству операций с приоритетной очередью, что в свою очередь ускоряет выполнение алгоритма.

1.5 Все алгоритмы






Все алгоритмы, разреженный граф
Вершины
Ребра
1e9
Дейкстра
Флойд-Уоршелл
Беллман-Форд
A*
Время, нс
Количество вершин
Количество ребер
0
200
400
600
800
1000
1
2
3
4
5
6
7
8

Вывод

По результатам измерений можно сказать, что лучшими алгоритмами для поиска кратчайшего пути между двумя вершинами являются алгоритм Дейкстры и алгоритм A*. Если необходимо найти кратчайшие пути между всеми парами вершин, то тогда алгоритм Беллмана-Форда подойдет лучше, чем алгоритм Флойда-Уоршелла.