

С помощью кода на C++ были сгенерированы графы, подходящие под условие задачи.

Далее графы прогнал через 4 алгоритма:

1. Алгоритм Дейкстры
2. Алгоритм Белмана-Форда
3. Алгоритм Флойда-Уоршела
4. Алгоритм Джонсона

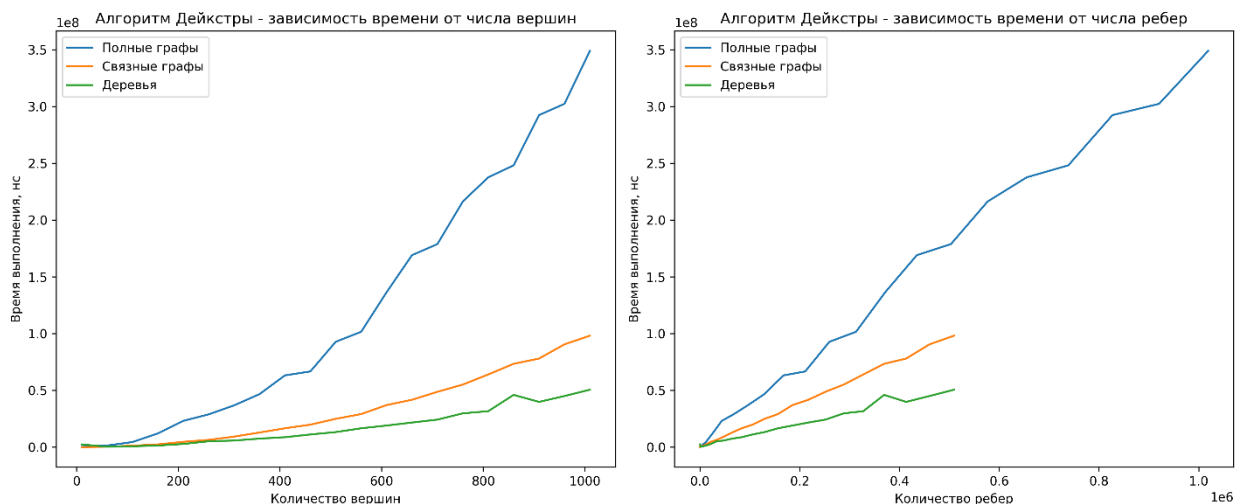
Время исполнения каждого алгоритма замерялось в наносекундах, затем данных были сохранены в csv таблицы, для каждого типа графа – своя таблица, в которой каждый из 4х алгоритмов на разном количестве вершин и ребёр, и соответствующее время выполнения алгоритма.

Затем, используя полученных таблицы строим графики на python:

1. Для каждого алгоритма зависимость времени от числа вершин и зависимость времени от числа ребёр.
2. Агрегированные графики для каждого типа графов (отдельно для вершин и для ребёр) – для сравнения эффективности алгоритмов.

## Результаты и выводы

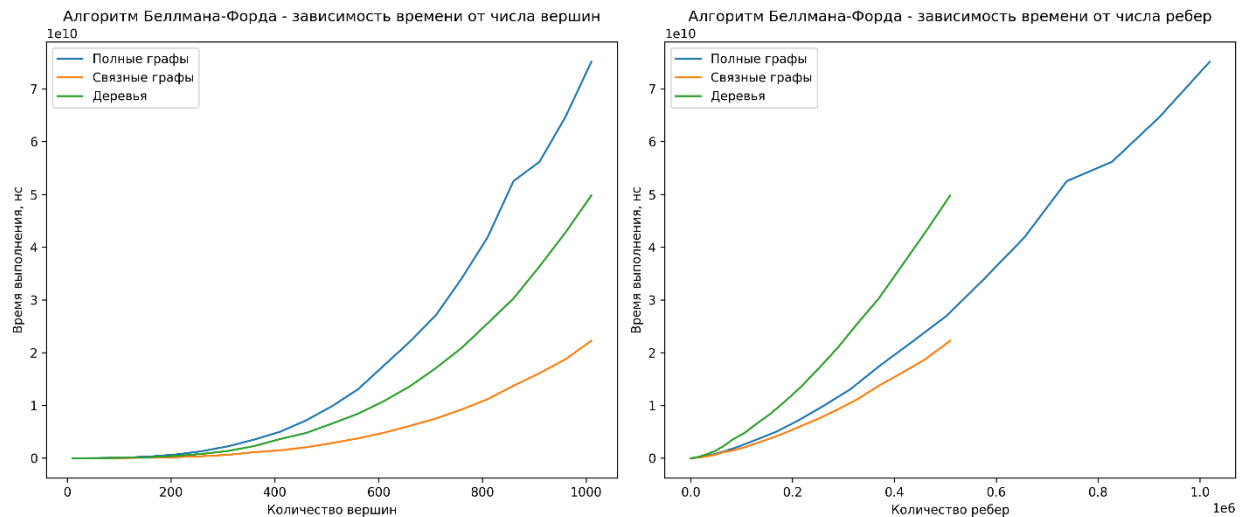
### Алгоритм Дейкстры



Можно сделать вывод, что при равном количестве вершин алгоритм работает дольше в зависимости от количества ребёр между этими вершинами. Это видно как по левому графику зависимости, так и по правому. Поскольку в полных графах все вершины связаны между собой, то алгоритм работает медленнее, чем, например, на деревьях.

Итого всего работает на графах деревьях, хуже всего на полных графах.

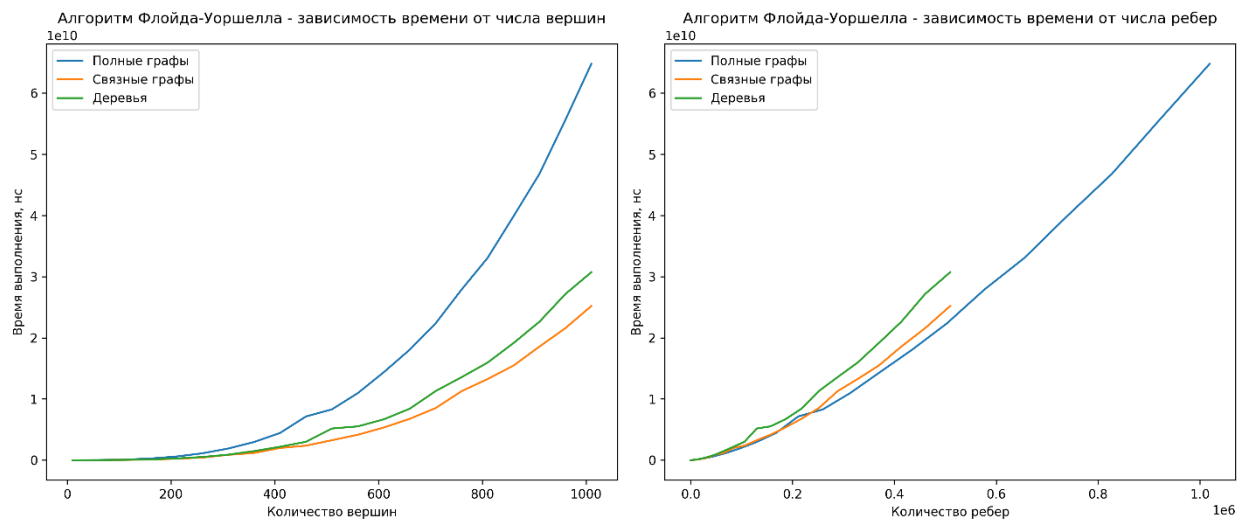
## Алгоритм Беллмана-Форда



Можно наблюдать похожую ситуацию, чем больше ребёр, тем хуже работает алгоритм. Однако Беллман-Форд лучше работает для связанных графов, чем для деревьев.

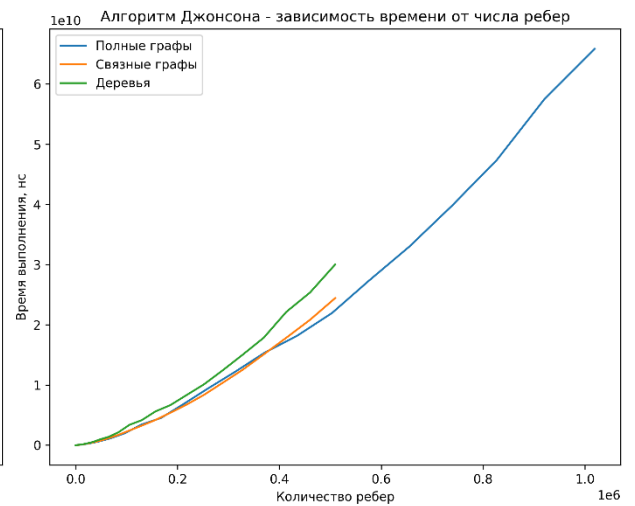
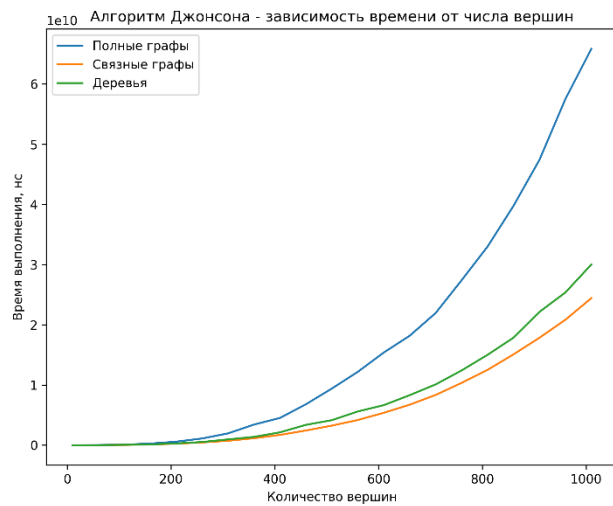
Итого всего работает на связанных графах, хуже всего на полных графах.

## Алгоритм Флойда-Уоршелла



Здесь можно наблюдать интересную ситуацию, где от зависимости количества вершин для разных графов время выполнения сильно разнится. Но если взглянуть на количество ребёр, то алгоритм почти одинаково эффективен для полных и связанных графов.

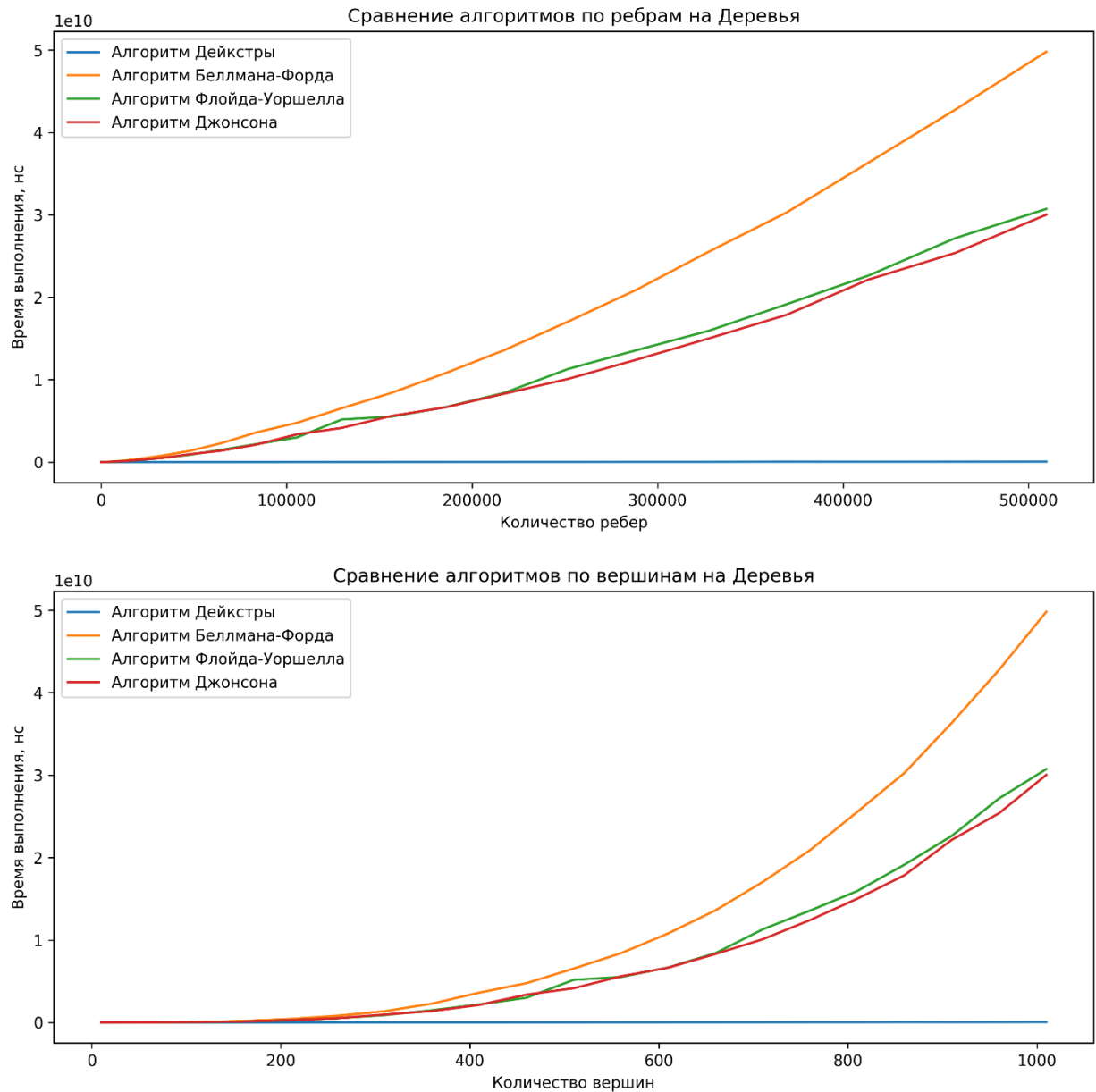
## Алгоритм Джонсона



Асимптотика почти одинакова для зависимости времени от числа ребёр графа. Графики очень похожи на алгоритм Флойда-Уоршелла.

## Общее сравнение алгоритмов

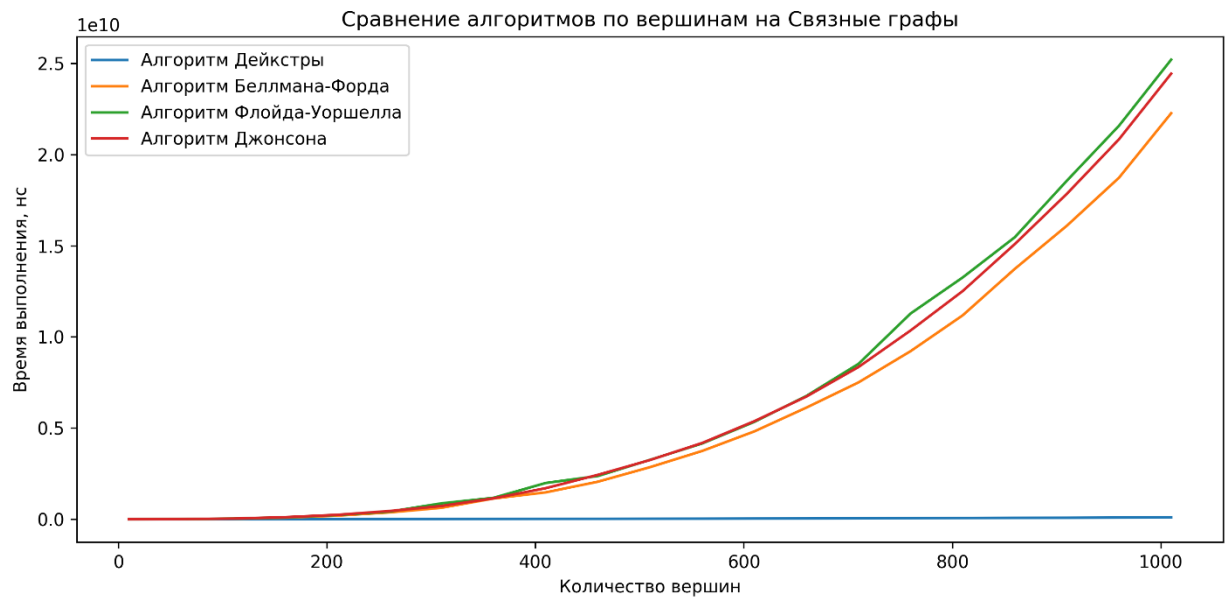
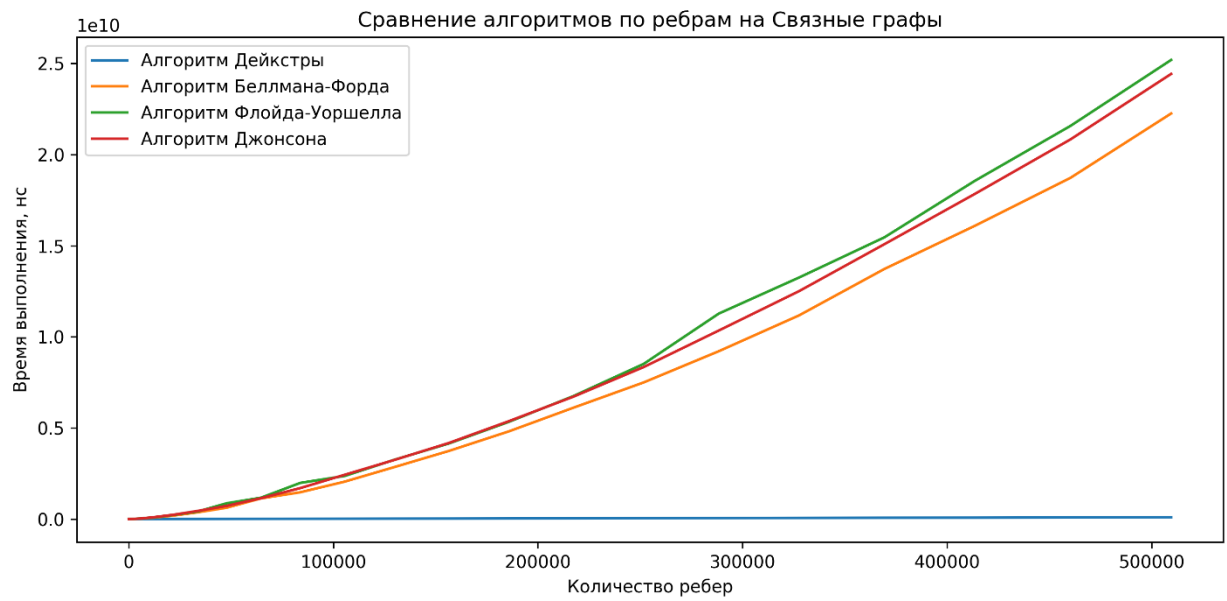
### Для деревьев



Очевидно, стоит использовать алгоритм Дейкстры при работе с деревьями.

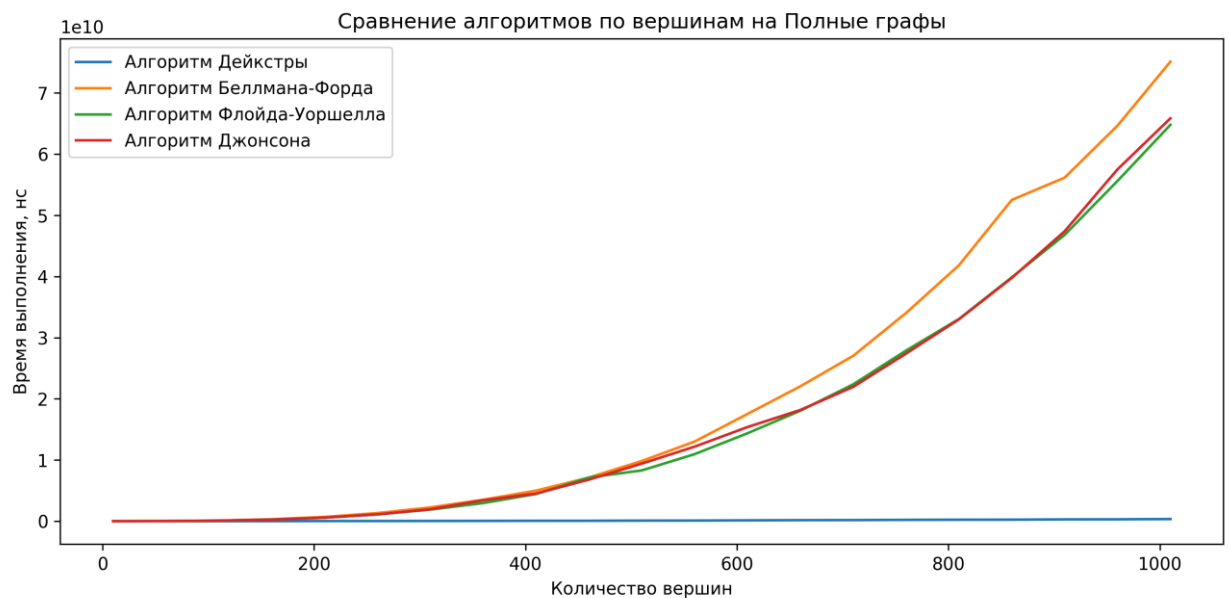
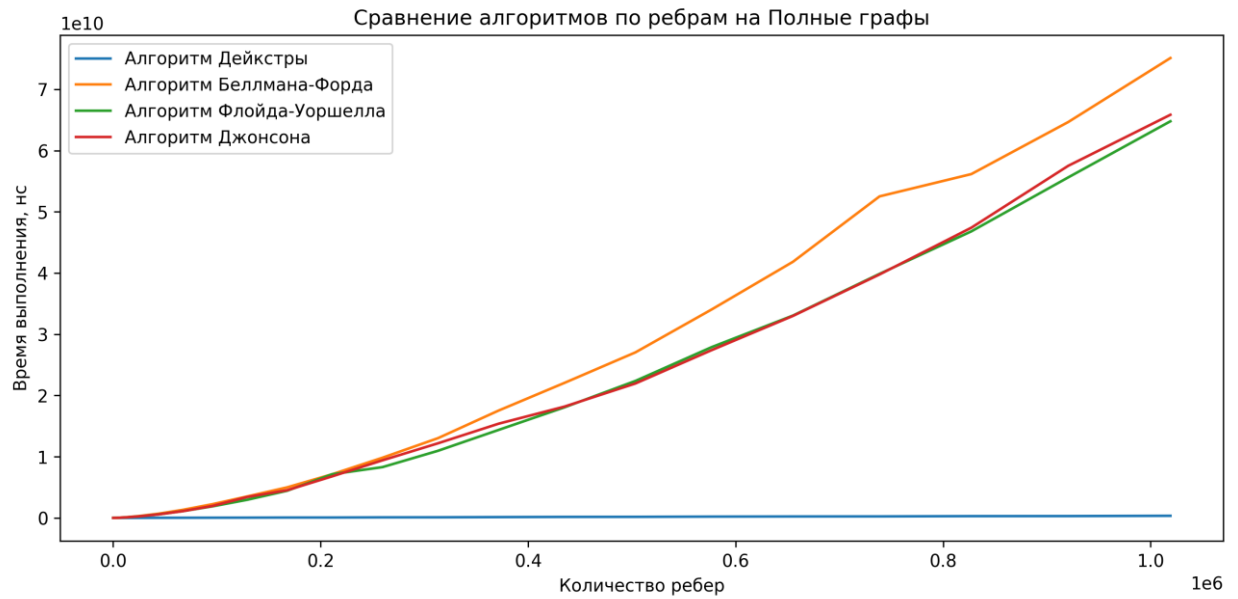
Хуже всех себя показал Беллман-Форд, а Флойд-Уоршелл и Джонсон примерно равны.

### Для связанных графов



Аналогично, лучше всех себя показывает алгоритм Дейкстры. Остальные примерно равно при работе со связанными графами.

## Для полных графов



И вновь, алгоритм Дейкстры лучше всех справляется с задачей.

Хуже всех алгоритм Беллмана-Форда

### Вывод

Лучше использовать алгоритм Дейкстры для любых типов графов!