

Постановка задачи

Сравнение алгоритмов и структур данных между собой.

Параметры вычислительного узла

- **GPU:** Intel Core i7-8700 3.20 GHz
- **RAM:** 16 GB
- **OS:** Windows 10 Home

Описание тестируемых алгоритмов

Будут тестироваться следующие алгоритмы поиска пути в графе:

- Поиск в глубину
- Поиск в ширину
- Дейкстра
- Беллман–Форд
- Флойд–Уоршелл

Поиск в глубину/в ширину являются базовыми алгоритмами поиска пути на графе.

Дейкстра позволяет нам находить кратчайшие пути от заданной вершины до всех остальных.

Беллман–Форд обладает функционалом Дейкстры, но также способен находить кратчайшие пути в графах с отрицательными весами(но без циклов отрицательного веса).

Флойд–Уоршелл позволяет найти кратчайшее расстояние между каждой парой вершин в графе.

Результаты измерений по времени и их анализ

Тест случайных данных без весов	Поиск в глубину	Поиск в ширину	Дейкстра	Беллман-Форд	Флойд-Уоршелл
100 вершин	0.0019791	0.0019859	0.0024808	0.2693521	0.1036687
200 вершин	0.0079359	0.0089276	0.0084318	2.47947	0.7817185
300 вершин	0.0243033	0.0218246	0.0223193	8.5440735	3.2869848
400 вершин	0.0406962	0.0381933	0.0391844	20.2104295	6.9970532
500 вершин	0.061995	0.0590486	0.0594928	42.0304283	14.4593519

Как видно из таблицы, **на графе без весов** лучше всего показывают себя поиск в глубину, поиск в ширину и Дейкстра, у них рост затраченного времени линейный по отношению к размеру входных данных.

Беллман-Форд же оптимальным не является, т.к. рост времени его работы экспоненциальный.

Сразу упомяну, что Флойд-Уоршелл выполняет немного другую задачу, т.к. он ищет кратчайшие пути между **каждой парой** вершин, следовательно и времени ему всегда понадобится больше, чем, например, Дейкстре.

Поэтому немного некорректно сравнивать его с остальными алгоритмами, но приведу результаты его работы на тех же входных данных ради статистической справки.

Тест случайных данных с неотрицательными весами	Поиск в глубину	Поиск в ширину	Дейкстра	Беллман-Форд	Флойд-Уоршелл
100 вершин	0.0019819	0.0019861	0.0019832	0.3080478	0.1036633
200 вершин	0.0079355	0.0084604	0.0089287	2.5648085	0.8903174
300 вершин	0.0208066	0.0203356	0.0208561	8.4692028	2.9586378
400 вершин	0.0381918	0.0372001	0.0376961	21.3933892	7.3809807
500 вершин	0.0624687	0.0669612	0.0605366	42.1296527	14.8779519

На графе с неотрицательными весами по-прежнему лучше всего себя показывают поиск в глубину/в ширину и Дейкстра, Беллман-Форд же тратит слишком много времени на большом размере входных данных.

Итого, для поиска кратчайшего пути на графе без весов или на графе с неотрицательными весами лучше всего использовать либо поиск в глубину, либо поиск в ширину, либо Дейкстру.

Применение алгоритма Беллмана-Форда оправдано только в том случае, если в графе присутствуют отрицательные веса, т.к. в этом случае подойдет только он.

Для поиска же кратчайшего расстояния между каждой парой вершин рекомендуется использовать алгоритм Флойда-Уоршелла.

Для каждого алгоритма проводилось **по три запуска**, здесь показано **среднее время** работы алгоритмов по итогам этих запусков:

Среднее время данных с неотриц. весами	Поиск в глубину	Поиск в ширину	Дейкстра	Беллман-Форд	Фloyd-Уоршелл
100 вершин	0.001993133333	0.001975466667	0.003141466667	0.2982782667	0.1124163667
200 вершин	0.008607066667	0.008919266667	0.003202644444	2.4826381	0.8369233667
300 вершин	0.0271226	0.02710433333	0.02761086667	8.4542729	2.853471167
400 вершин	0.03984716667	0.03821063333	0.03818213333	19.88638777	6.7889101
500 вершин	0.0644712	0.06018913333	0.07820216667	39.95358337	13.41145613

Также рассчитано **стандартное отклонение**:

Стандартное отклонение данных с неотриц. весами	Поиск в глубину	Поиск в ширину	Дейкстра	Беллман-Форд	Фloyd-Уоршелл
100 вершин	9.46E-11	8.48E-11	1.01E-06	7.16E-05	5.75E-05
200 вершин	3.38E-07	1.58E-07	2.46E-05	0.005063980977	0.002138192094
300 вершин	2.99E-05	3.44E-05	3.42E-05	0.0001671764355	0.008295015522
400 вершин	2.06E-06	7.66E-07	1.77E-07	1.703289983	0.2629106965
500 вершин	3.01E-06	3.44E-05	0.0002340541843	3.551458297	1.612957383

Результаты измерений используемой памяти и их анализ

По результатам **двух запусков**:

Используемая память в килобайтах	Поиск в глубину	Поиск в ширину	Дейкстра	Беллман-Форд	Фloyd-Уоршелл
100 вершин	16	16	12	291	579
200 вершин	23	24	23	2997	4176
300 вершин	23	24	23	3440	6203
400 вершин	62	63	46	7506	12432
500 вершин	62	63	46	8262	16087

Из результатов можно заметить, что дейкстра потребляет меньше всего памяти, поиск в ширину и поиск в глубину идут примерно наравне и потребляют памяти чуть больше, чем Дейкстра.

Алгоритм Беллмана-Форда же потребляет существенно больше памяти, чем вышеперечисленные алгоритмы, из-за особенностей его реализации (динамическое программирование).

Алгоритм Флойда-Уоршелла, ожидаемо, потребляет больше всех памяти из-за своей реализации через матрицу смежности.