

Отчёт КДЗ 3

Краткое описание кода:

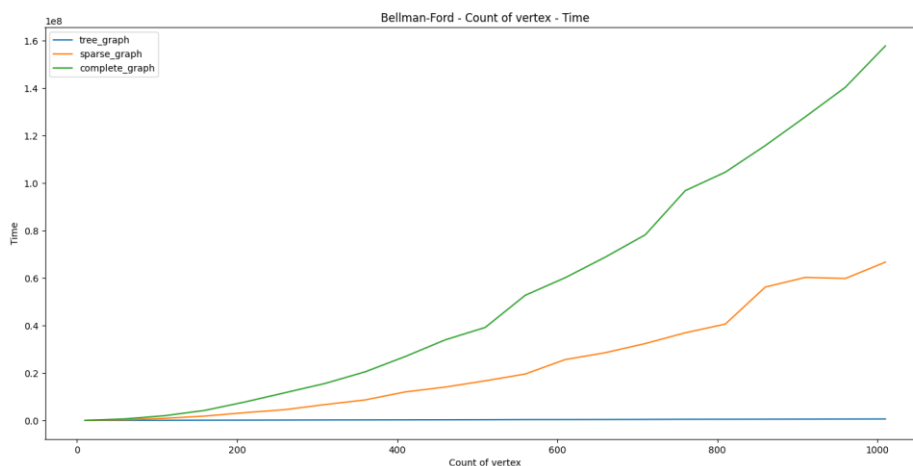
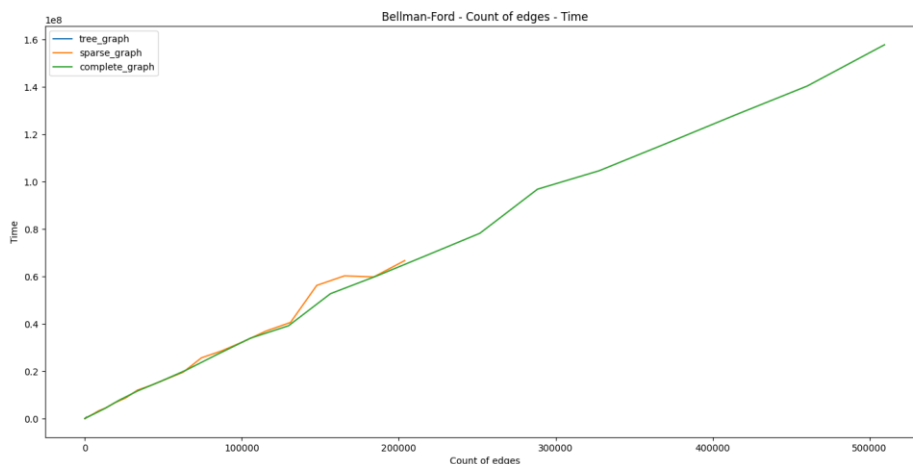
В файле **main.c** находится основной код программы. В начале программы инициализируется массив пар **searchers** (название функции, ссылка на функцию), далее массив передается в функцию **printAndAnalyse**, где в цикле вызывается каждая функция и тестируется на разных типах и размерах графов (по количеству вершин ребер). В функции **analyse** реализован подсчет усредненного времени работы алгоритма.

Алгоритмы: Дейкстры, Флойда-Уоршела, Форда-Беллмана реализованы в соответствующих функциях: `dijkstra`, `BellmanFord`, `FloydWarshall`.

В папке `graphs` находятся графики с результатами тестов алгоритмов.

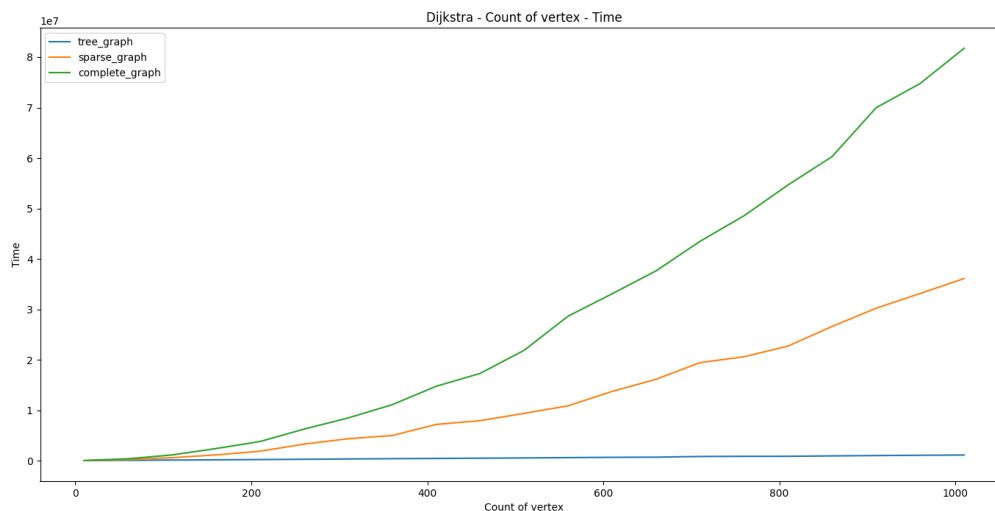
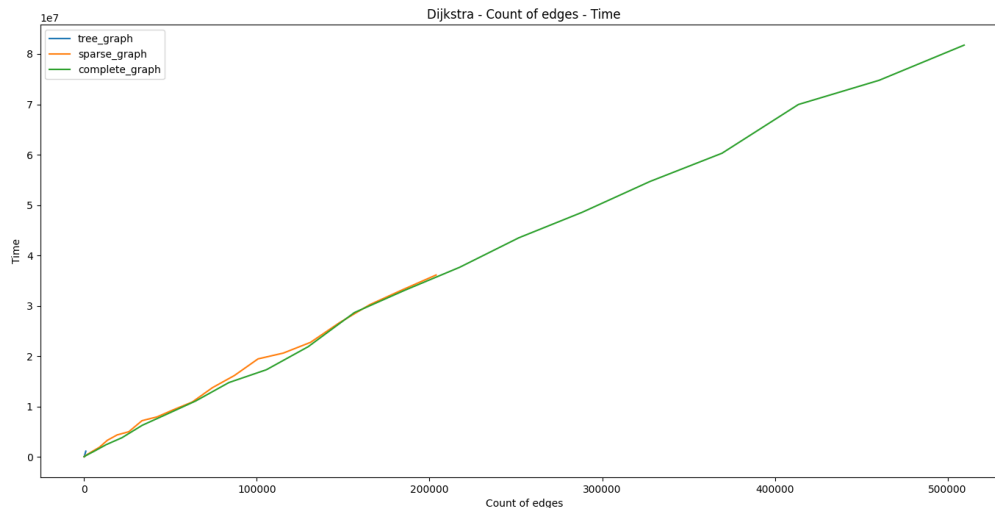
Анализ алгоритмов:

1. Алгоритм Форда-Беллмана



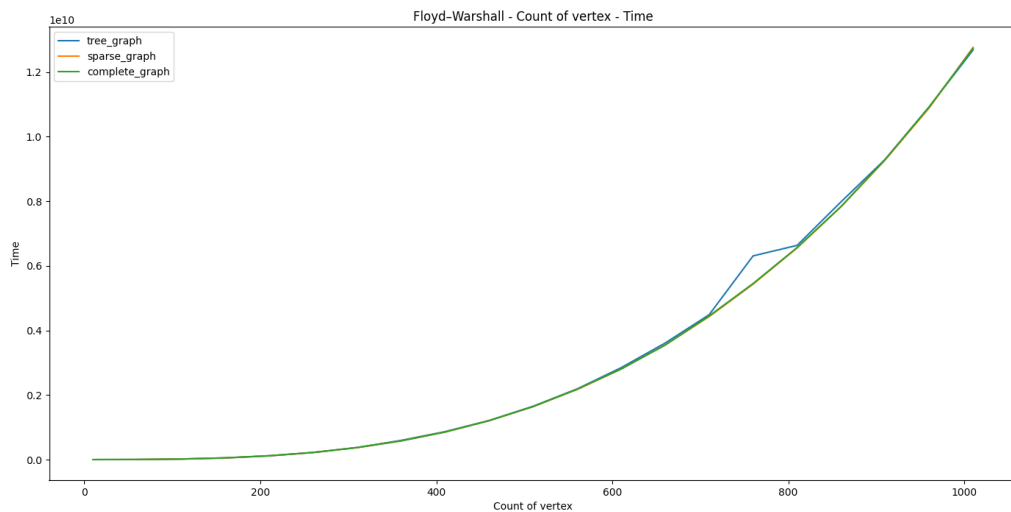
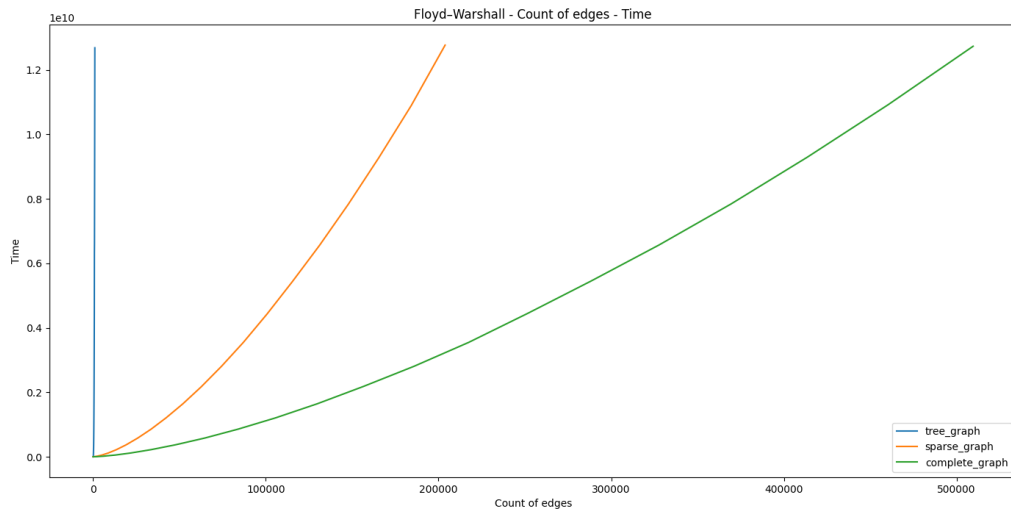
Алгоритм производит поиск кратчайшего пути от одной вершины до всех остальных и имеет асимптотическую сложность равную $O(m * n)$. На данных графиках можно увидеть, что сложность работы алгоритма асимптотически совпадает с графиком времени, которое зависит на первом графике от количества ребер, на втором от количества вершин. Погрешности на первом графике могут создаваться из-за того, что алгоритм не имеет фиксированного шага по количеству ребер у разных типов графов.

2. Алгоритм Дейкстры



Алгоритм производит поиск кратчайшего пути от одной вершины до всех остальных. Алгоритм реализован через **set** и имеет асимптотическую сложность равную $O(m * \log n)$. Из графиков видно, что асимптотически зависимость времени от количества вершин и ребер имеют схожую сложность со сложностью работы алгоритма. Погрешности на первом графике могут создаваться из-за того, что алгоритм не имеет фиксированного шага по количеству ребер у разных типов графов.

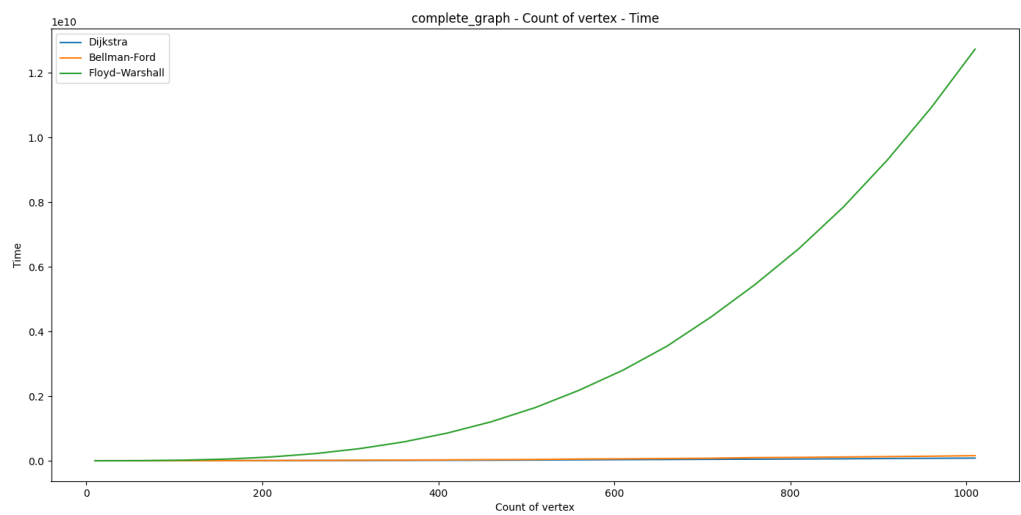
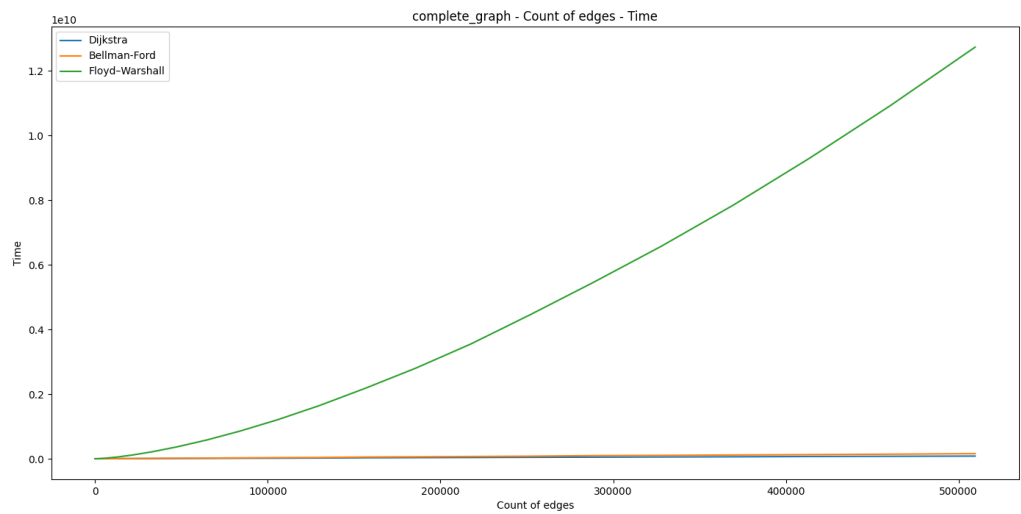
3. Алгоритм Флойда-Уоршелла



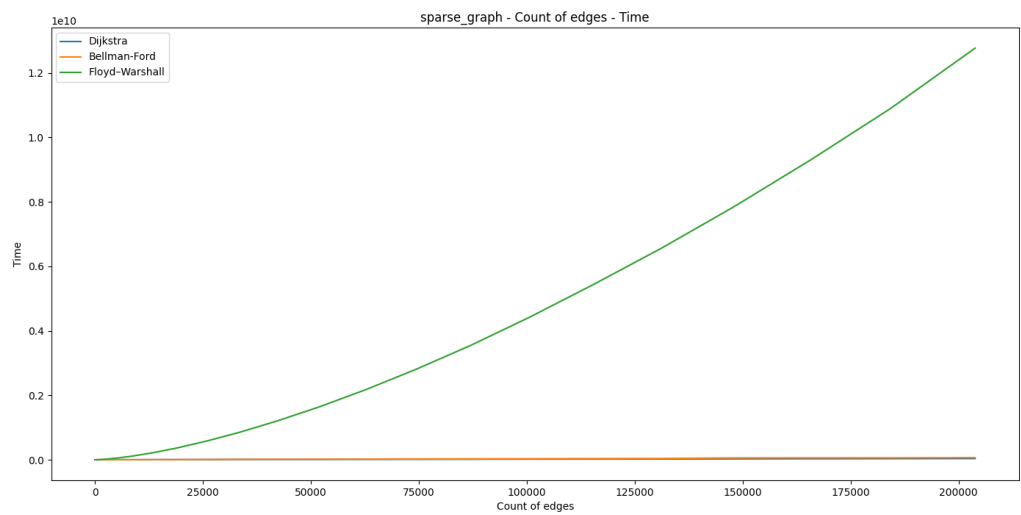
Динамический алгоритм, который производит поиск кратчайших путей для каждой из вершин графа. Имеет асимптотическую сложность $O(n^3)$. Из зависимости времени от количества вершин видим, что имеет все типы графов, работают примерно одинаково за кубическую сложность. На первом графике происходят такие расхождения, так как время работы алгоритма не зависит от количества ребер совершенно.

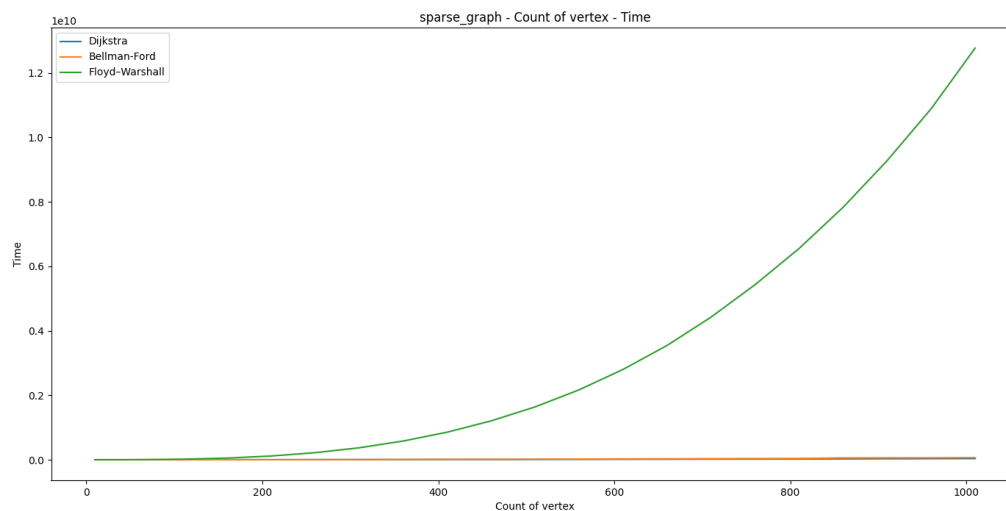
Все алгоритмы:

Полный граф

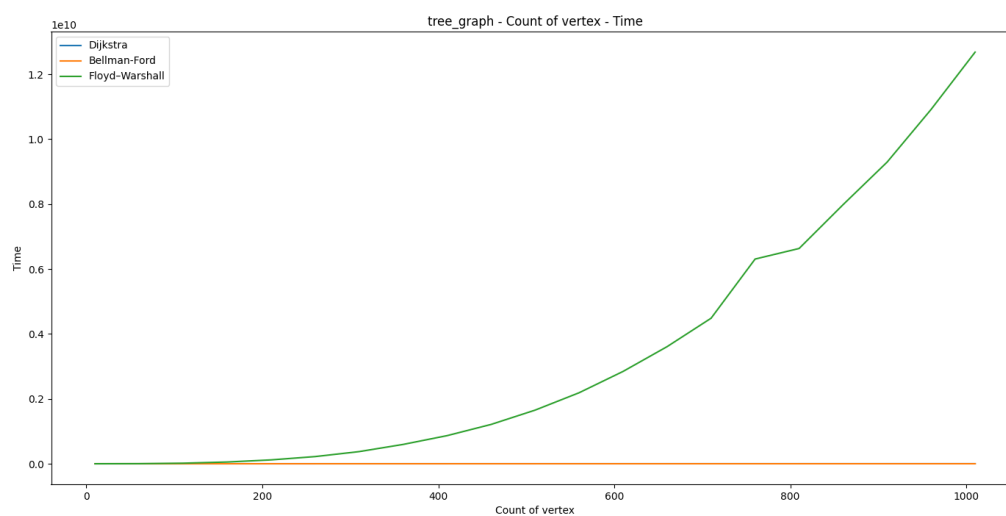
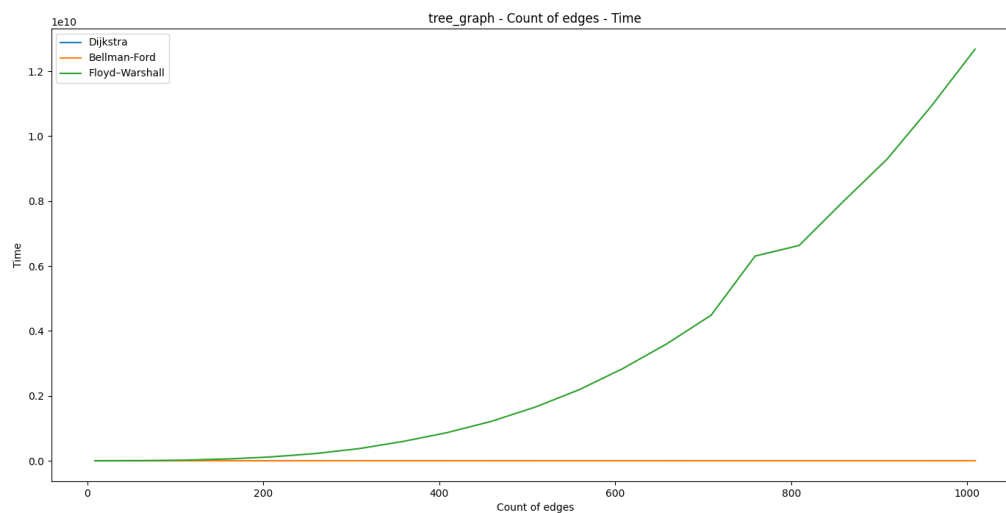


Связный граф





Разреженный граф



Из графиков зависимости времени работы от количества вершин и количества ребер заметно, что алгоритм Флойда-Уоршела работает хуже всего, имея

кубическую сложность работы, поэтому его необходимо использовать для поиска кратчайших путей для каждой из вершин графа, так же он имеет преимущество в том, что умеет работать с ребрами отрицательного веса.

Далее из графиков можно заметить, что на разреженном графе и на связном алгоритмы Дейкстры и Форда-Белмана работают с похожей асимптотикой, но на полном графе алгоритм Дейкстры выигрывает Форда-Белмана.