*Отчёт по задаче «Поиск в графе»*

**Постановка задачи**

Задача заключается в реализации библиотеки, предоставляющей алгоритмы поиска пути в графе (орграфе), и в ее последующем тестировании по производительности и по потреблению памяти.

**Параметры вычислительного узла**

Процессор: Ryzen 5 2600, 6 ядер, 12 потоков

Память: DDR4 2x8 3400MHz, CL17

ОС: Windows 10 Professional 64bit

**Тестируемые алгоритмы**

1. Алгоритм Беллмана-Форда

Алгоритм поиска кратчайшего пути во взвешенном графе. За время O(|V|\*|E|) {\displaystyle O(|V|\cdot |E|)}алгоритм находит кратчайшие пути от одной вершины графа до всех остальных. Допускает рёбра с отрицательным весом

1. Алгоритм Дейкстры

Алгоритм на графах, который находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса

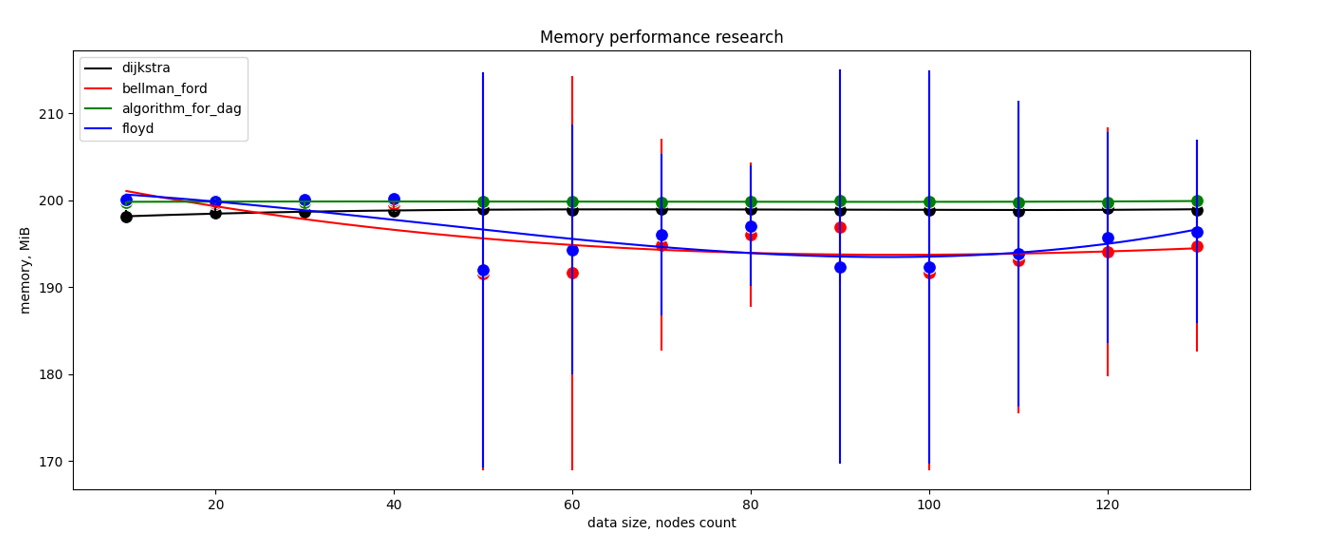
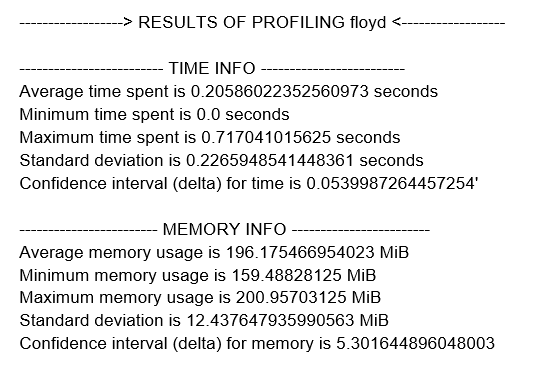
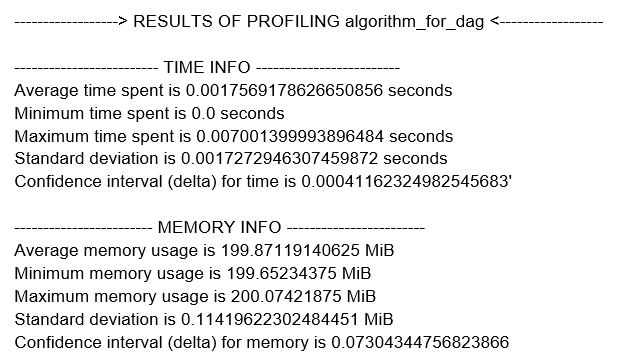
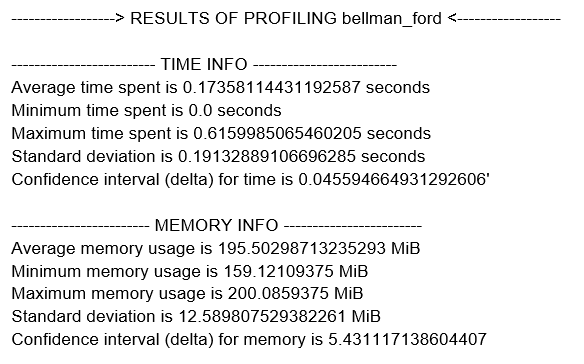
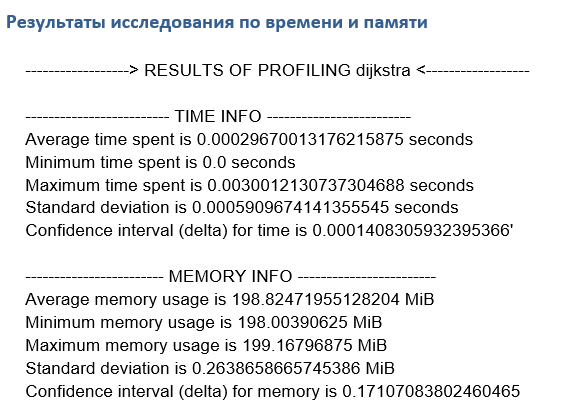
1. Алгоритм Флойда

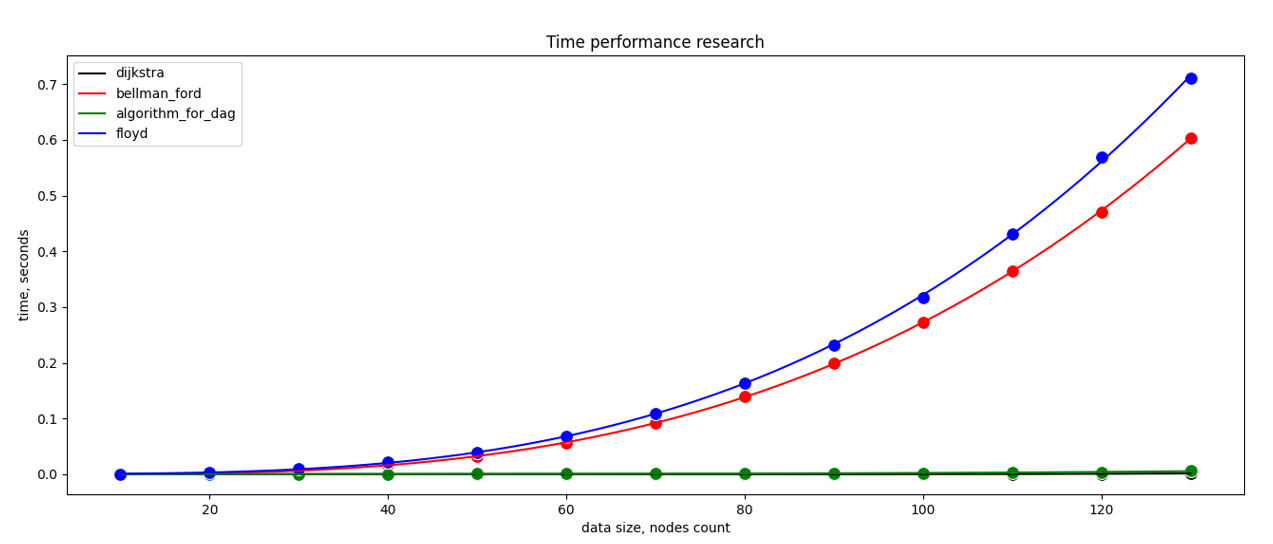
Алгоритм поиска кратчайших путей во взвешенном графе с положительным или отрицательным весом ребер (но без отрицательных циклов). За одно выполнение алгоритма будут найдены длины (суммарные веса) кратчайших путей между всеми парами вершин

1. «Безымянный» алгоритм

Алгоритм поиска кратчайших путей от одной до всех вершин во взвешенном ацикличном графе

**Результаты измерений**





**Обоснование результатов**

Все алгоритмы тратят примерно одно количество памяти, несмотря на то что алгоритм Флойда должен потреблять больше памяти, так как результат алгоритма хранит пути между всеми вершинами, в отличие от остальных алгоритмов, которые хранят путь от одной вершины до всех остальных. По времени выполнения алгоритм Дейкстры и «Безымянный» алгоритм лидируют, так как их время выполнения O(n^2). Алгоритм Флойда и алгоритм Беллмана-Форда имеют похожее время выполнения. Алгоритм Флойда имеет временную сложность O(n^3), а алгоритм Беллмана-Форда имеет сложность O(|V|\*|E|), где V – множество вершин, E – множество рёбер, но в случае генерации данных почти все графы имеют n вершин и n^2 рёбер, из чего и получается, что алгоритм Беллмана-Форда схож по сложности с алгоритмом Флойда, так как |V| = n, |E| ≈ n^2. Стандартное отклонение значительно больше у алгоритмов, которые имеют временную сложность O(n^3), так как при росте размера входных данных скорость выполнения значительно увеличивается.

**Анализ результатов**

Результаты показывают, что предпочтительным алгоритмом для поиска пути от одной вершины до всех в графах без отрицательных весов является алгоритм Дейкстры, так как он эффективен не только по временной сложности, но и по памяти. Если в графе нет цикла, то предпочтительным вариантом будет алгоритм для бесконтурных сетей (algorithm\_for\_dag, «Безымянный алгоритм»), потому что он может работать в графах с отрицательными весами, и он практически не отстаёт в эффективности от алгоритма Дейкстры. И если в графе есть и отрицательные веса, и циклы, то выбор падает на алгоритм Беллмана-Форда. В графах, где количество вершин примерно равно количеству рёбер данный алгоритм не будет отставать от алгоритма Дейкстры и алгоритма для бесконтурных сетей по скорости и по потреблению памяти.