

Отчет по проведенному эксперименту

Лунев Артем

Информация об оборудовании:

Процессор: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2712 МГц, ядер: 2, оперативная память 8 Гб, DDR4 (для docker было выделено 6 Гб).

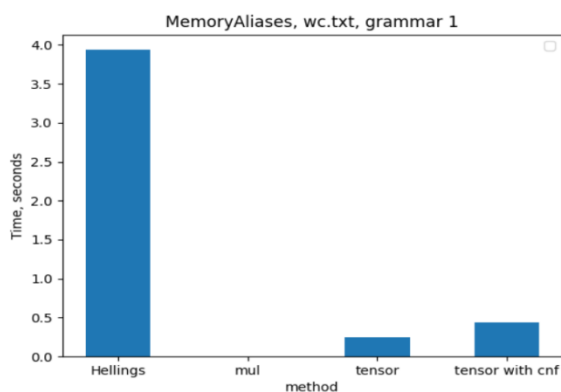
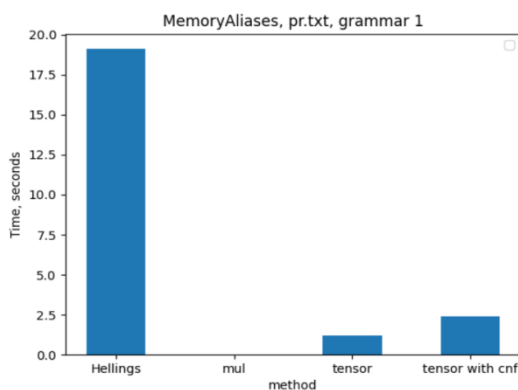
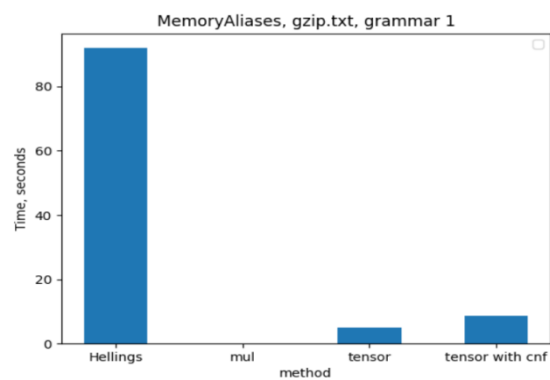
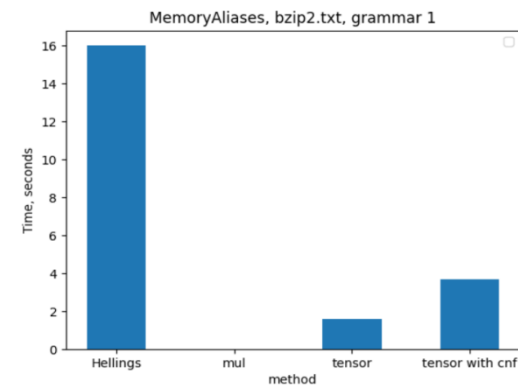
Эксперимент:

Эксперимент проводился на графах 3 видов: Full graph, Memory aliases, Worst case (на Sparse graph не удалось дождаться конца вычислений).

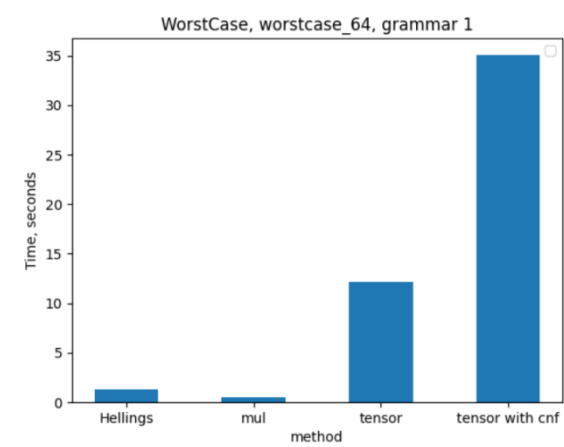
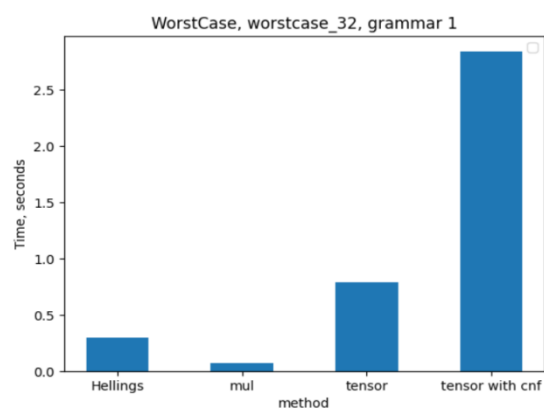
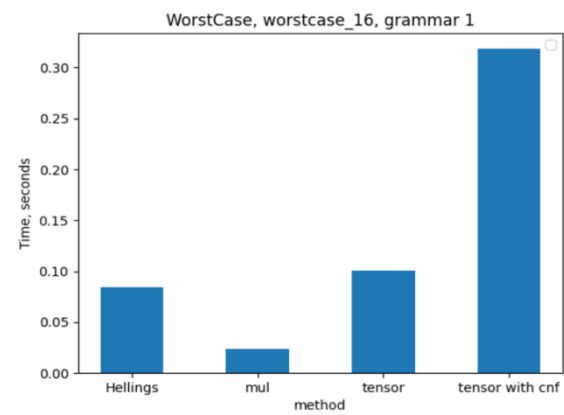
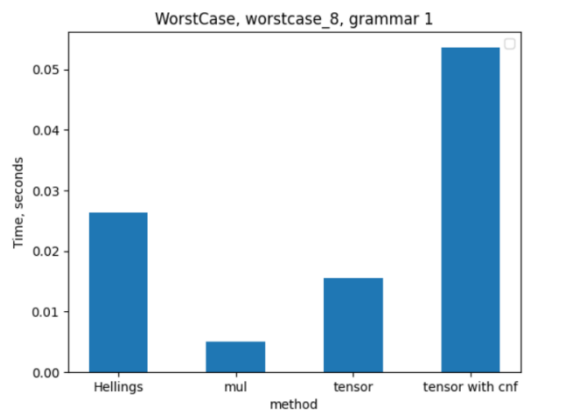
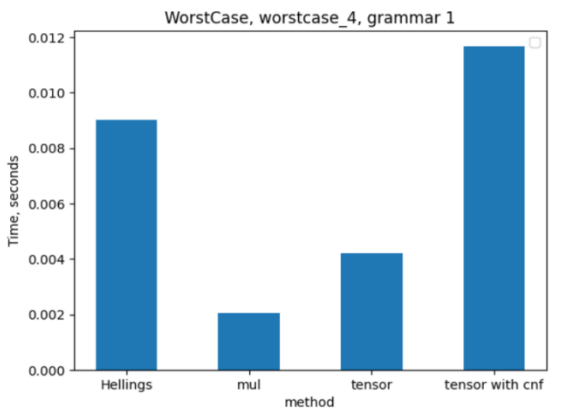
Все замеры проводились 3 раз с использованием библиотеки time, затем бралось среднее.

Ниже приведены результаты замеров времени для 3 видов графов с различными грамматиками.

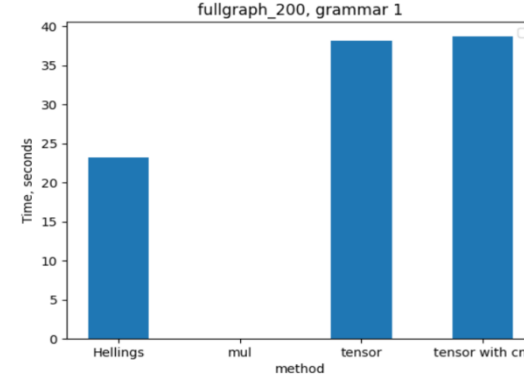
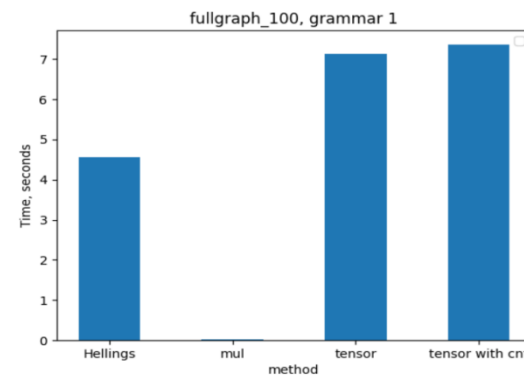
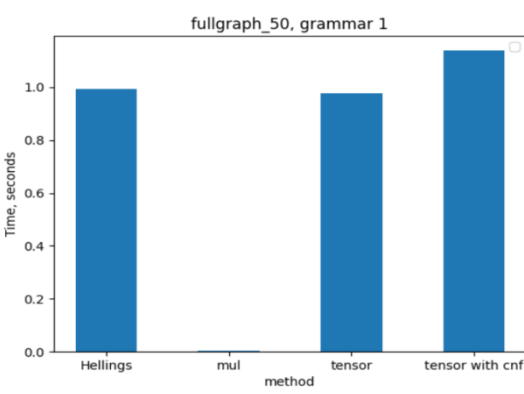
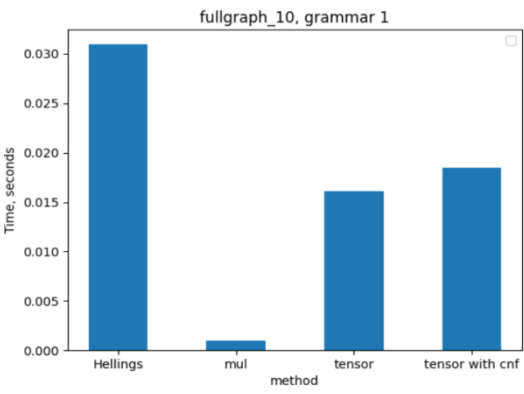
Memory aliases



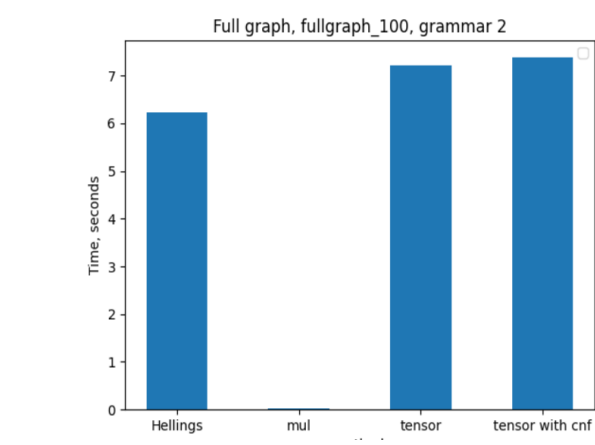
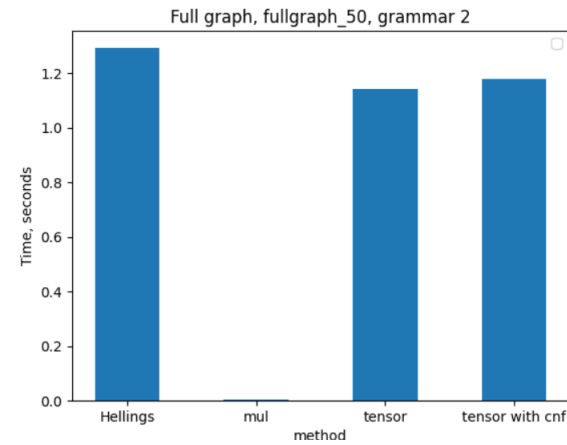
Worst case



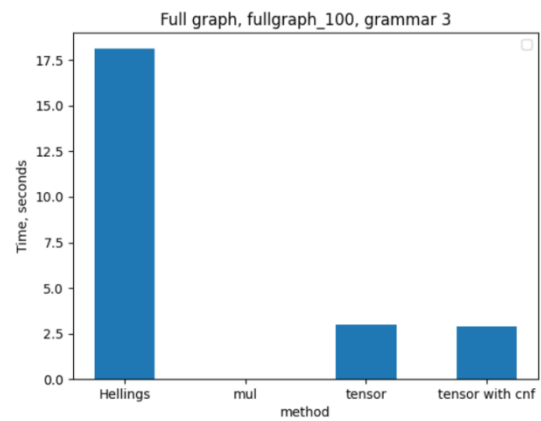
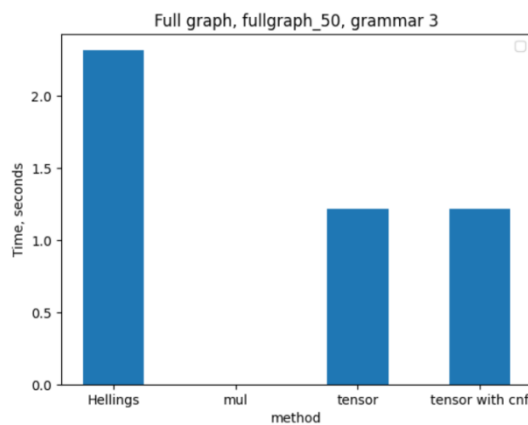
Full graph (gr 1)



Full graph (gr 2)



Full graph (gr 3)



Вывод:

Как видно из графиков с результатами вычислений алгоритм, основанный на перемножении матриц показывает лучший результат на всех замерах. При сравнении тензорного алгоритма и алгоритма Хеллингса оказывается, что в большинстве случаев тензорный работает быстрее. В некоторых случаях тензорный алгоритм все же проигрывает, но показывает более стабильный результат, в то время как время работы Хеллингса бывает крайне большим (в том числе в Sparse graph, там его не удалось дожидаться).

Также следует заметить, что тензорный алгоритм, принимающий грамматику в ОНФХ работает медленнее, чем он же без преобразования грамматики, из-за того, что в первом случае размер грамматики больше чем во втором.