# Задача A3 отчёт

## Ссылка на публичный репозиторий:

https://github.com/samstarkov/set3-algo/

**set_a3i.cpp** – код, который засылался в codeforces, для проверки работоспособности алгоритма

ID посылки codeforces: 349344689

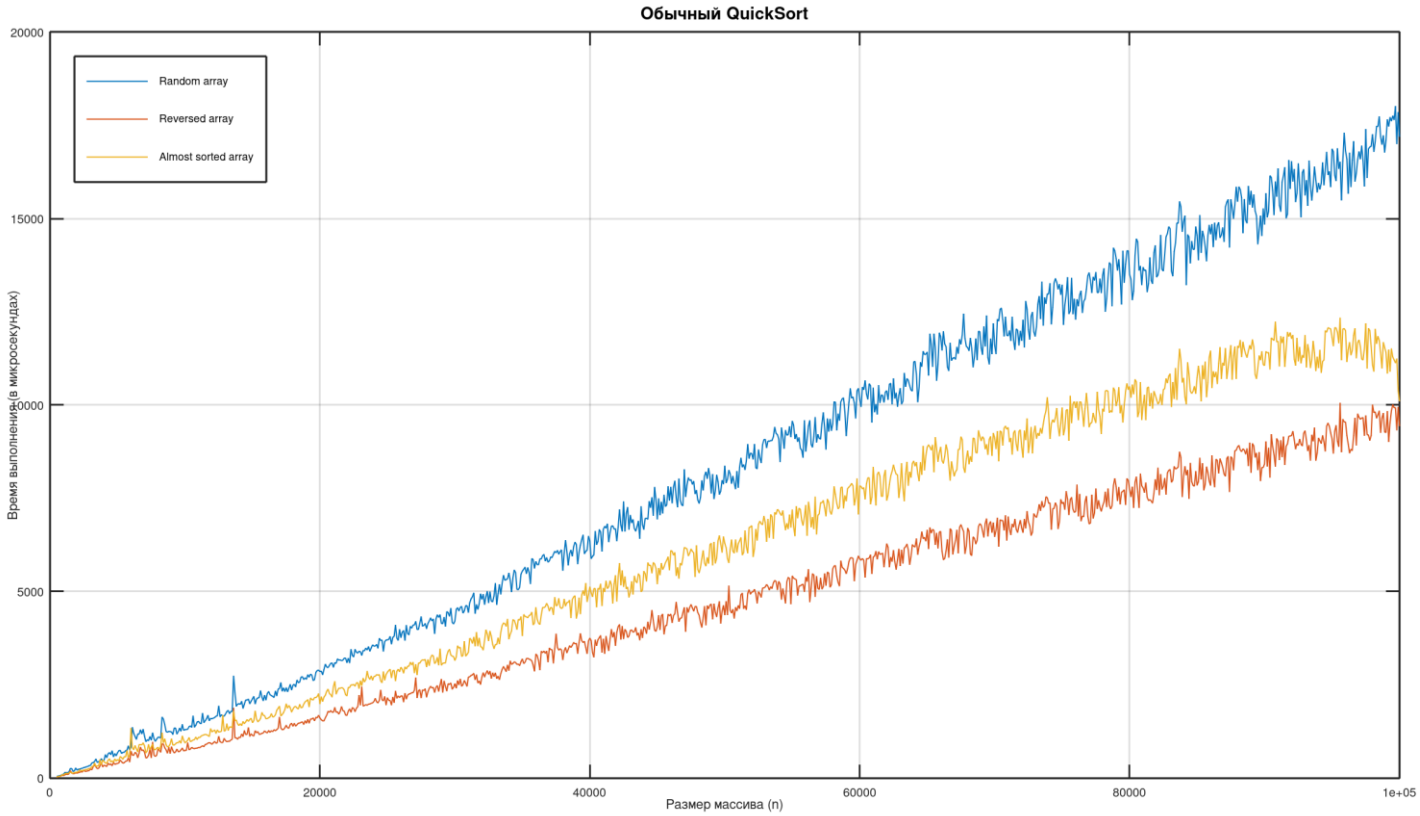
**set_a3_addition.cpp; SortFuncs.h; ArrayGenerator.h; SortTester.h;** - файлы содержащие код для получения данных для дальнейшего анализа (составления графиков).

**QuickSortRandom.txt; QuickSortReversed.txt; QuickSortAlmostSorted.txt;** - файлы с результатами обычного QuickSort для массивов различных видов. Каждый файл содержит два столбца, первый из которых – значения n, второй – результат замера времени (в микросекундах).

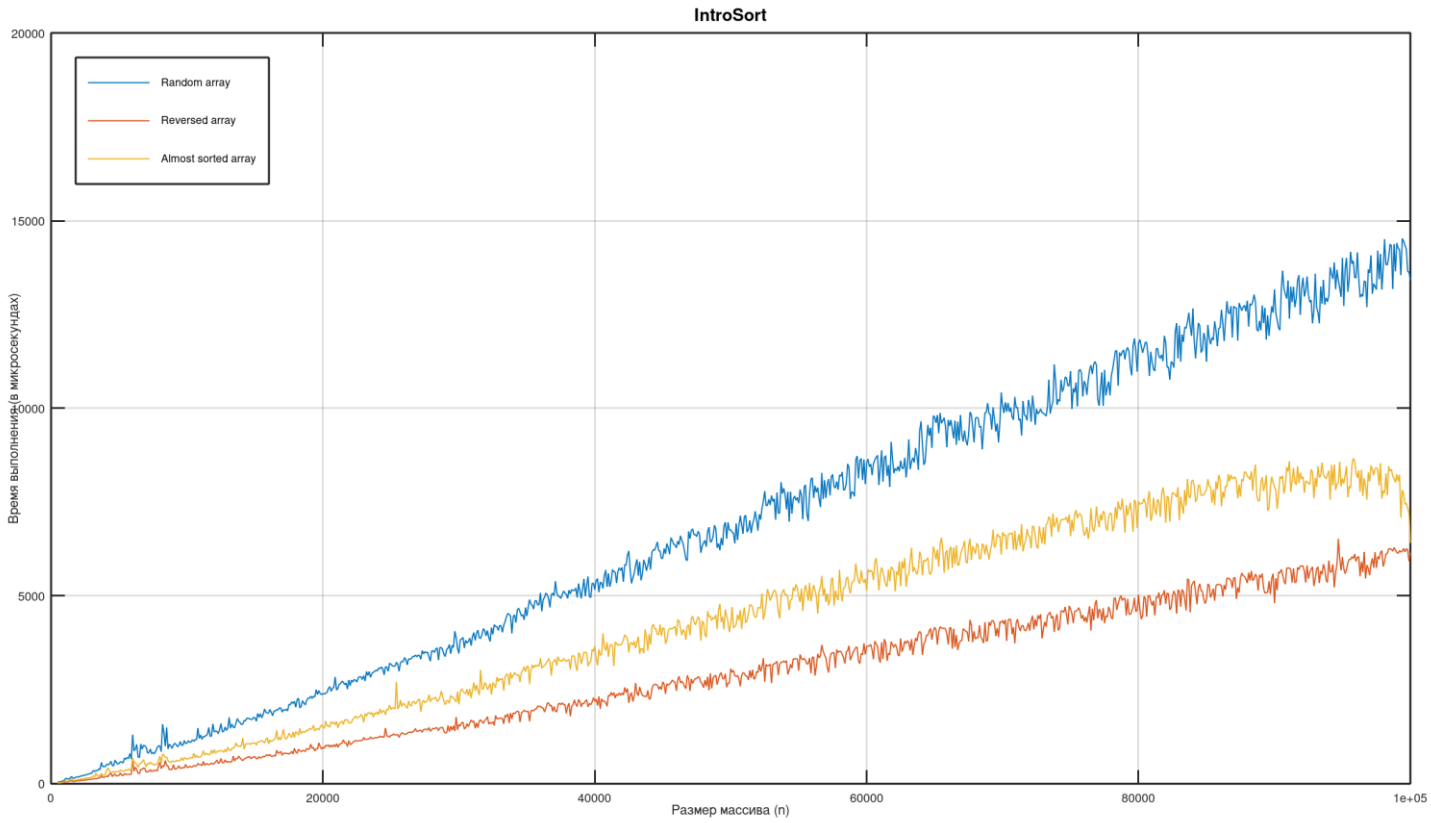
**IntroSortRandom.txt; IntroSortReversed.txt; IntroSortAlmostSorted.txt;** - файлы с результатами IntroSort для массивов различных видов. Каждый файл содержит два столбца, первый из которых – значения n, второй – результат замера времени (в микросекундах).

Далее, в Octave (MatLab) написан код, для представления графиков, необходимых по условию задачи (файл **set3_a3_plots.m**):

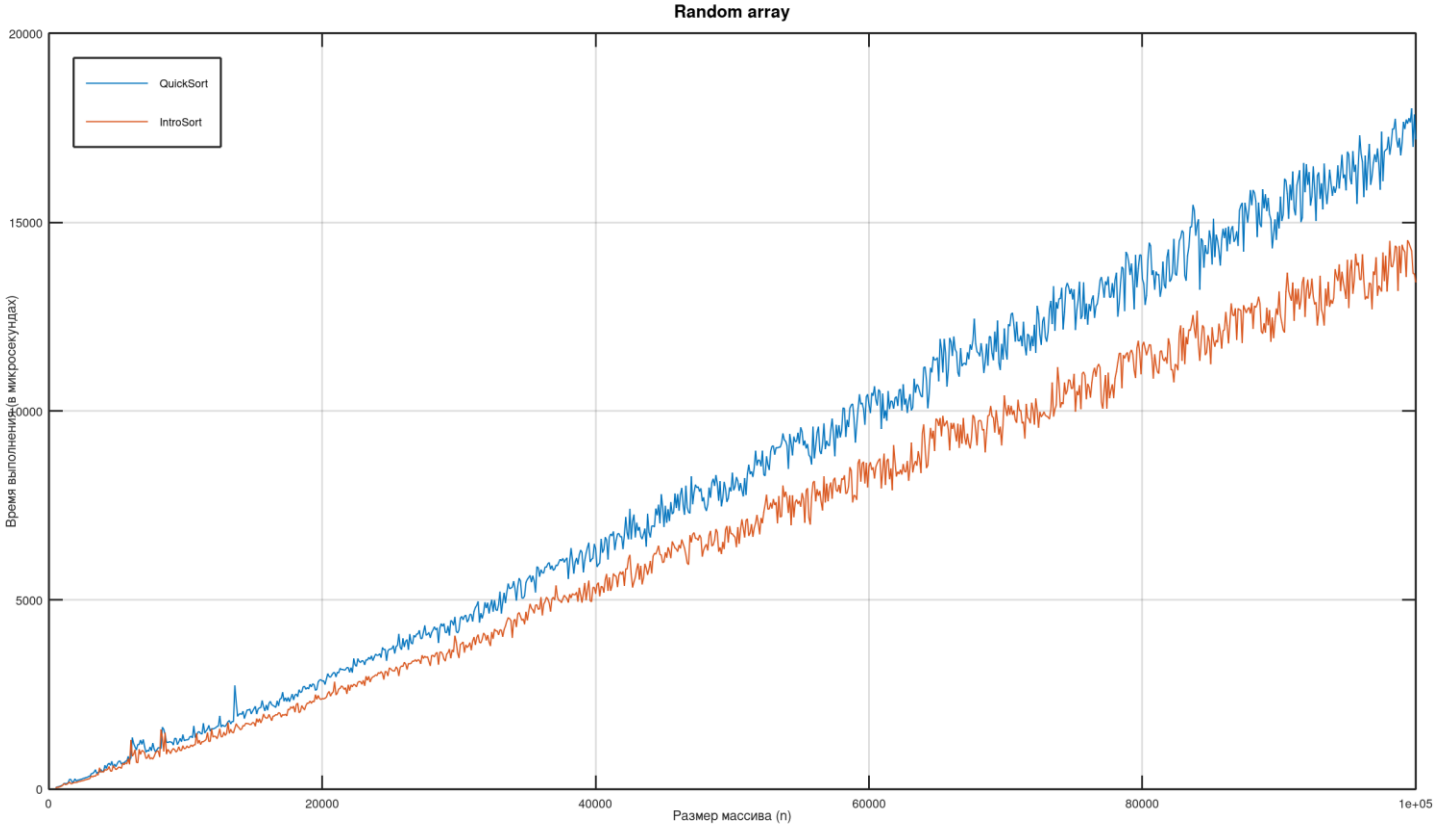
**Работа обычного QuickSort:**

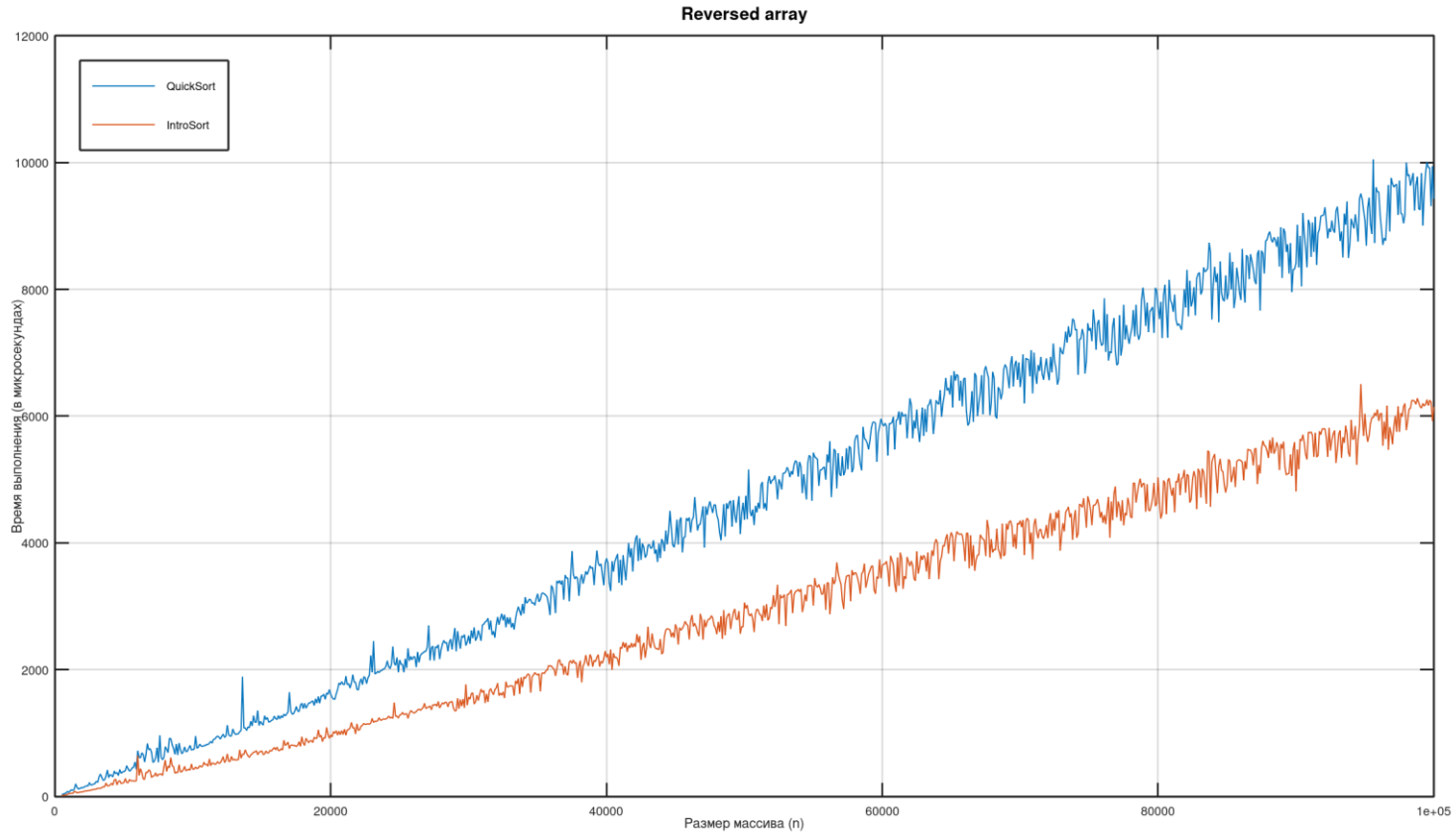


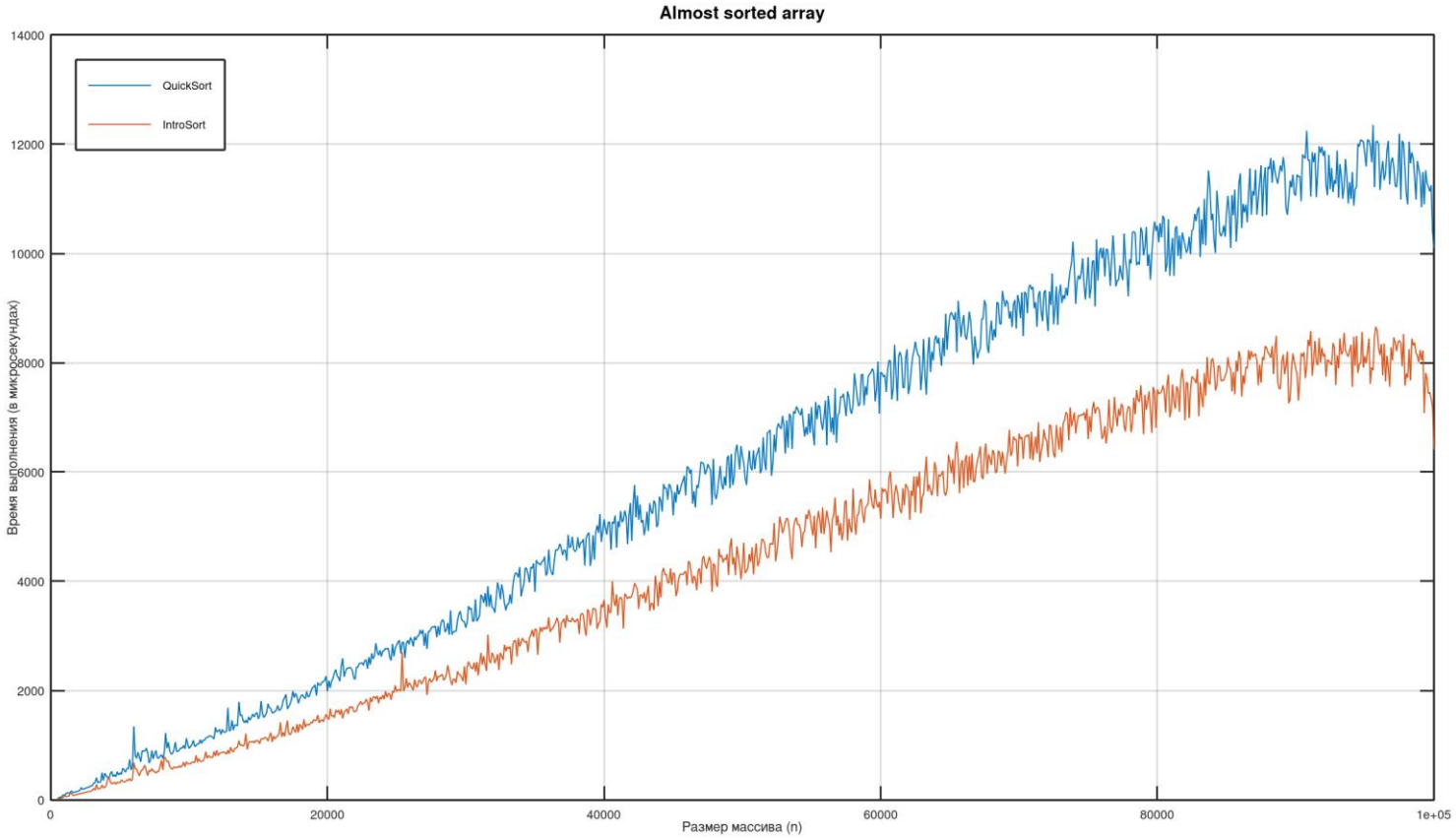
## Работа IntroSort:




## Сравнение эффективности обычного QuickSort и IntroSort:




Reversed array
QuickSort
IntroSort
Время выполнения (в микросекундах)
Размер массива (n)
0
2000
4000
6000
8000
10000
12000
0
20000
40000
60000
80000
1e+05


Almost sorted array
QuickSort
IntroSort
Время выполнения (в микросекундах)
Размер массива (n)
0
2000
4000
6000
8000
10000
12000
14000
0
20000
40000
60000
80000
1e+05

IntroSort показывает себя лучше во всех трёх сценариях, но наибольшая разница между QuickSort и IntroSort заметна на reversed массивах. Это вызвано переключением на HeapSort с гарантированной оценкой O(n*logn) при достижении определённого уровня (а затем в InsertionSort т. к. он хорошо работает для маленьких массивов), в то время как обычный QuickSort в наихудшем случае будет работать за O(n^2).

При этом сортировка, что для QuickSort, что для IntroSort, медленнее всего выполняется на random массивах, а быстрее всего на reversed массивах.

Исходя из полученных результатов, можно сказать, что использование IntroSort вместо QuickSort может являться хорошей оптимизацией.