

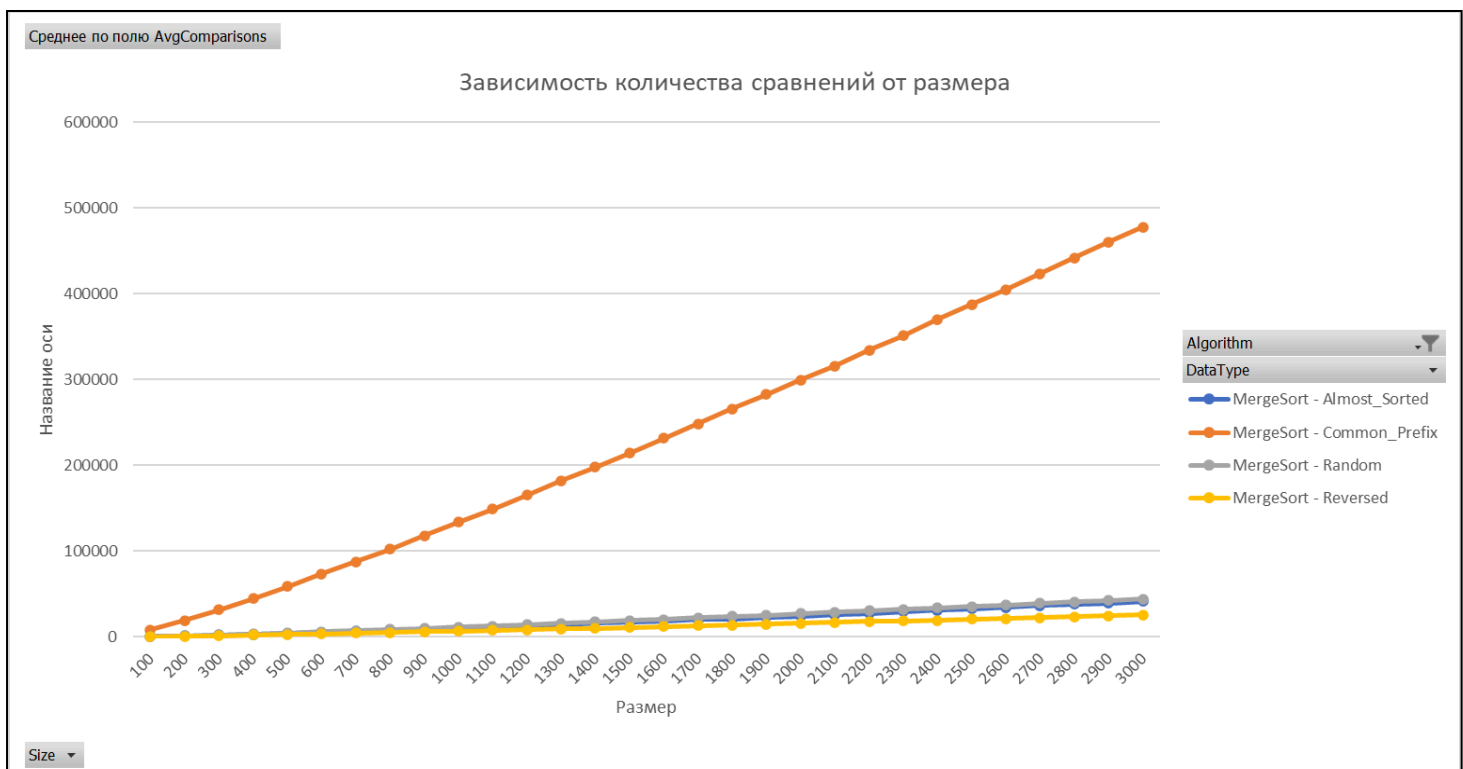
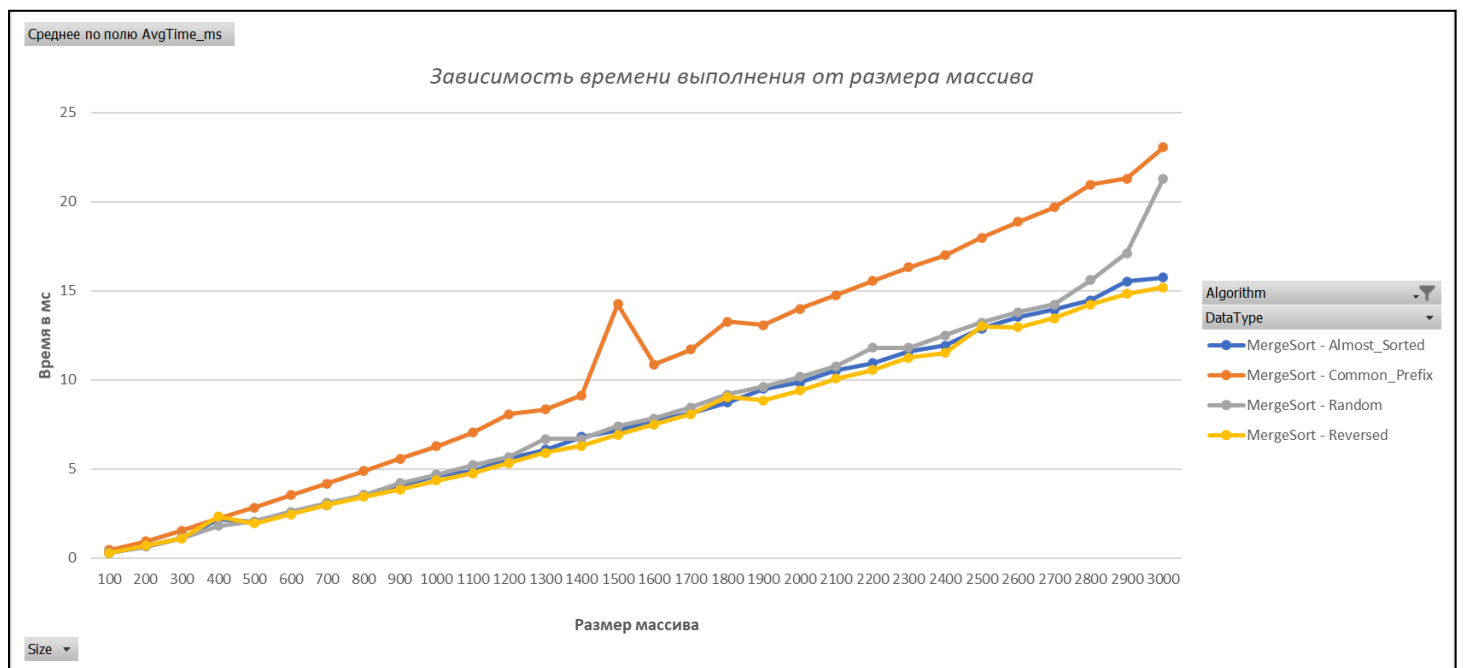
Тепляков Владислав БПИ2310

SET 9

Задача A1

MergeSort

Стабильная производительность: Время соответствует $O(n \cdot \log(n))$, Для $n = 3000$ AvgTime = 21.283 ms. На данных с общим префиксом наблюдается рост сравнений символов (до 477791 для $n=3000$), но время растет линейно-логарифмически.



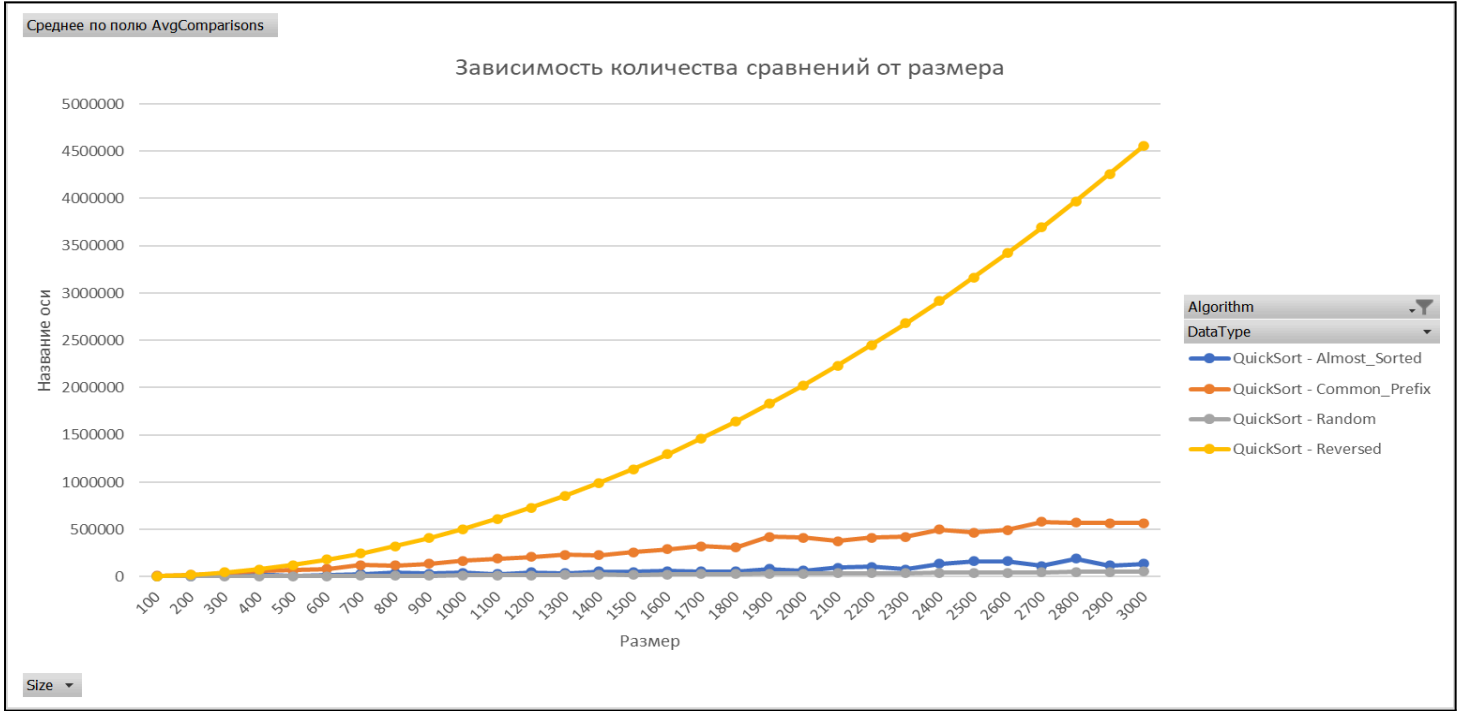
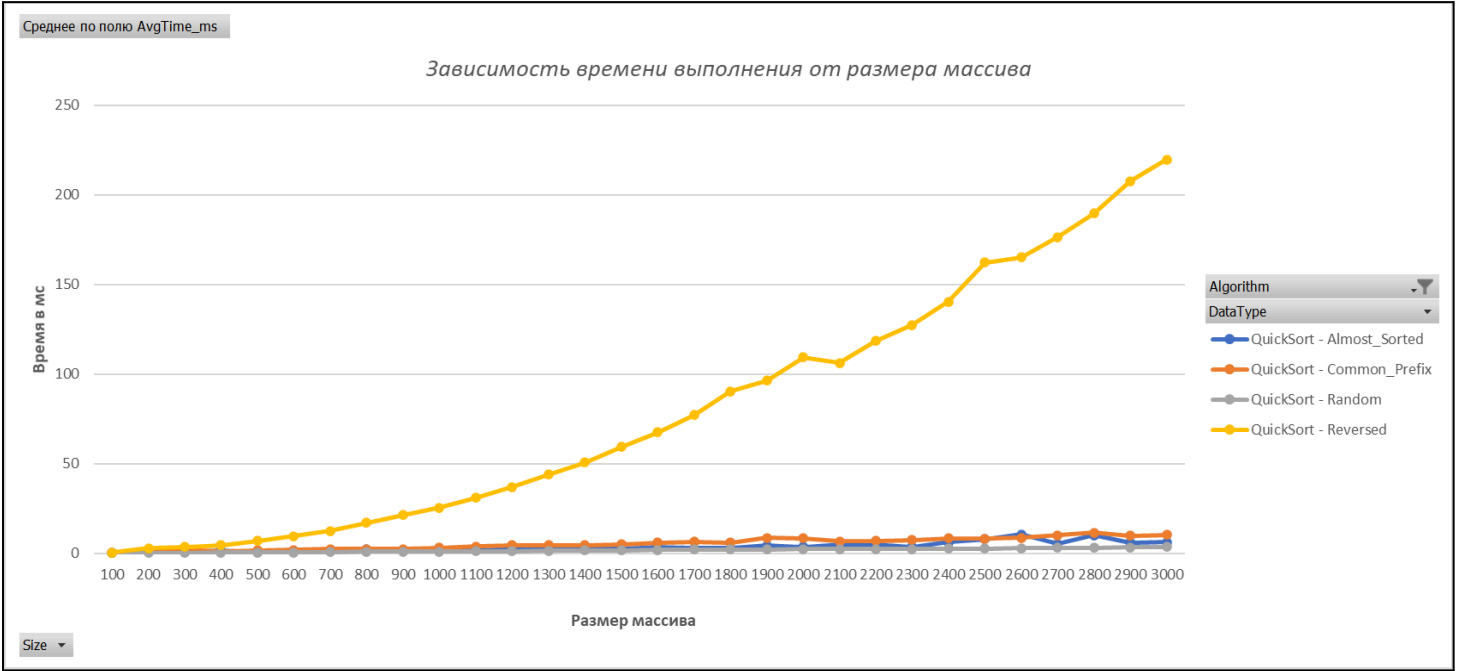
QuickSort

Случайные данные:

Время растёт пропорционально $O(n \log n)$. Например, для $n=3000$: AvgTime=3.542мс.
Количество сравнений: 14556 для $n=1000$.

Reversed/Common_Prefix:

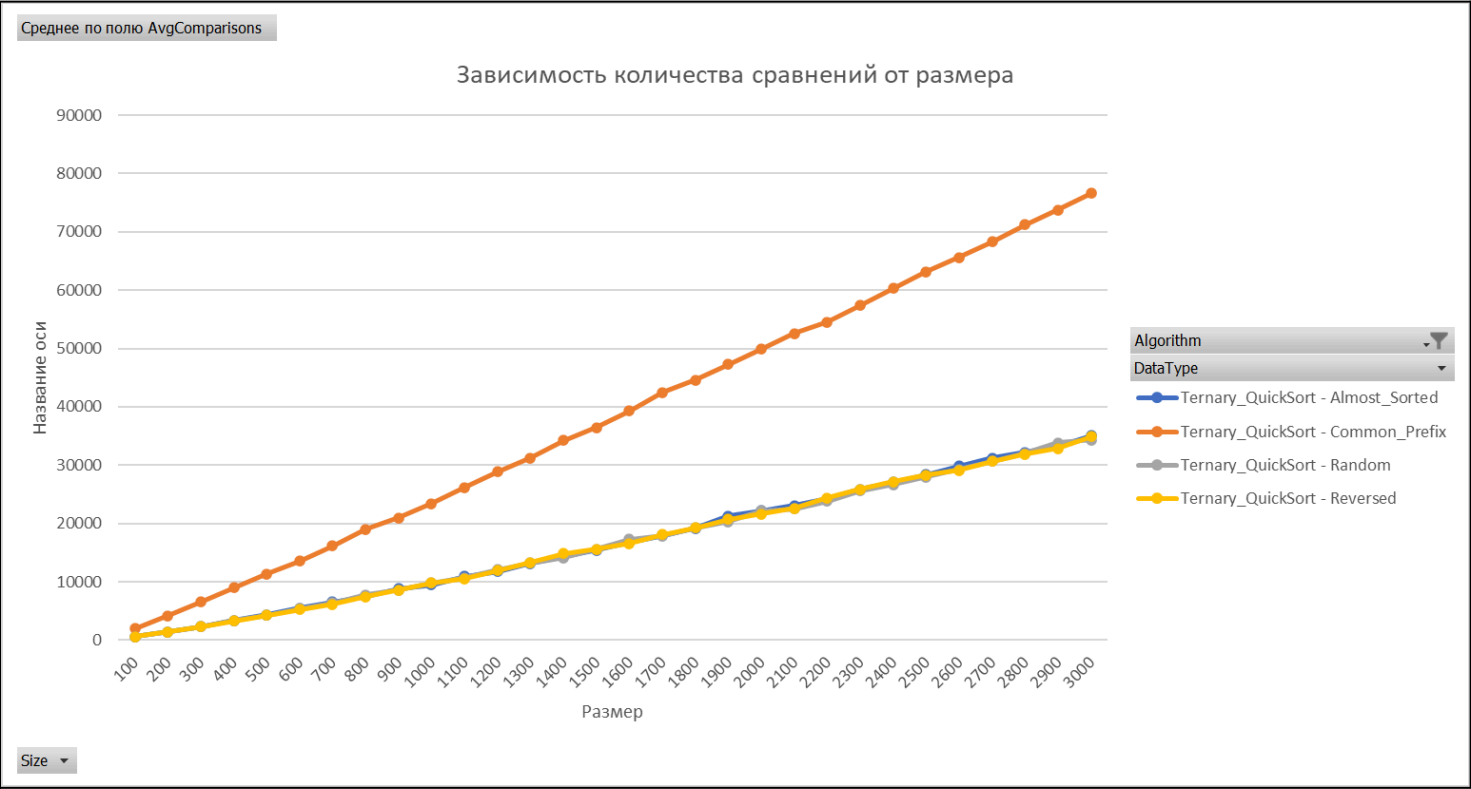
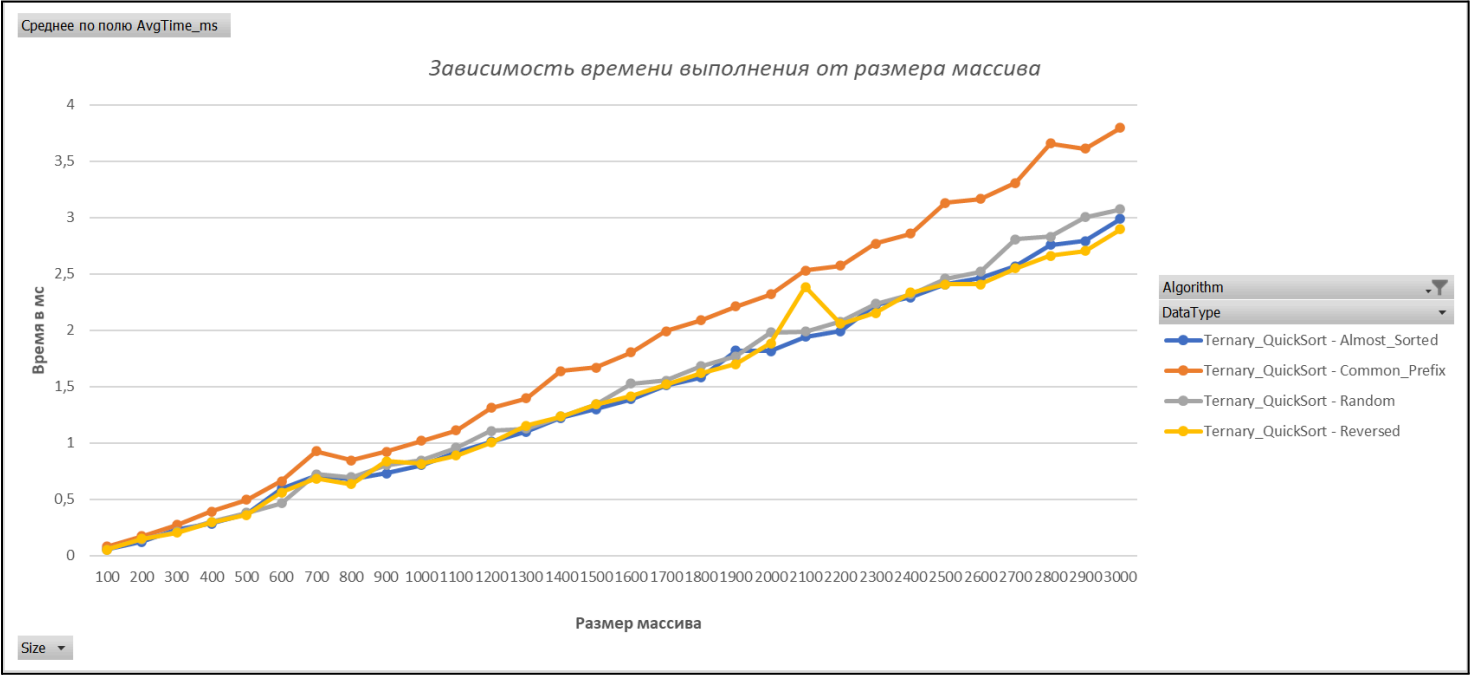
Деградация до $O(n^2)$: для $n=3000$ и Reversed — AvgTime=219.699мс. Высокие затраты на сравнение префиксов (для $n=1000$ и Common_Prefix — 171452 сравнений).



Ternary_QuickSort

Оптимизация для строк:

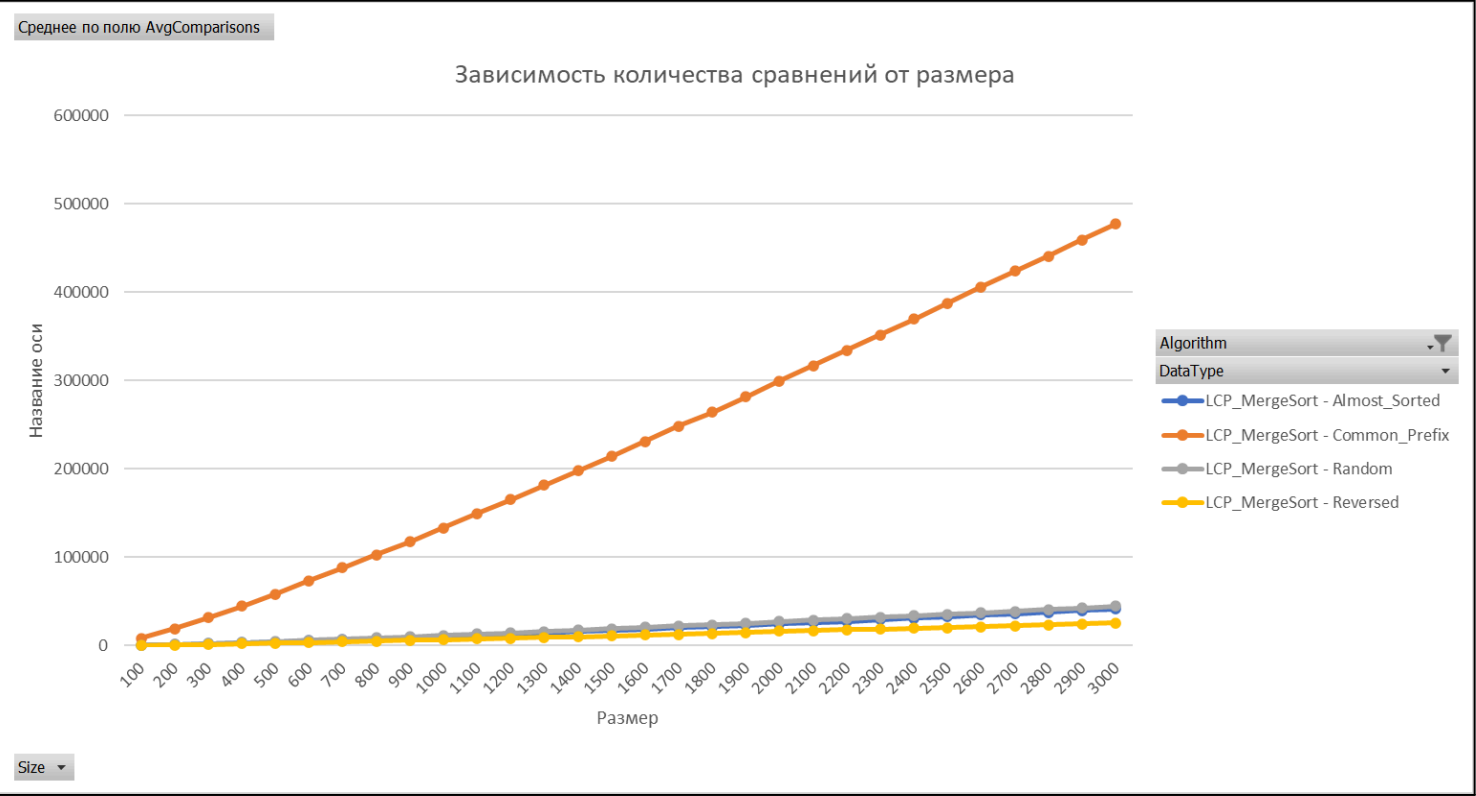
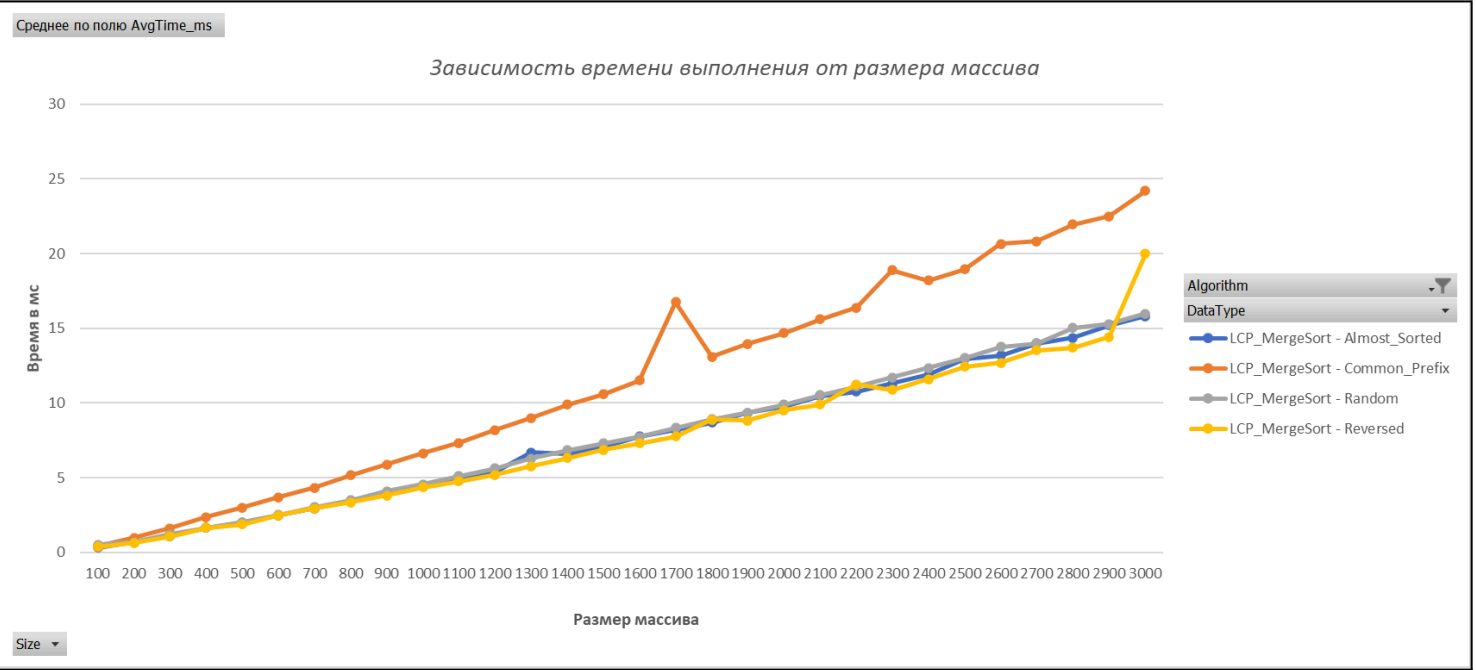
Лучшее время на всех типах данных. Для n=3000 и Random: AvgTime=3.073мс против 3.542мс у стандартного QuickSort. Снижение количества сравнений за счет тернарного разбиения (на 30% для Common_Prefix).



LCP_MergeSort

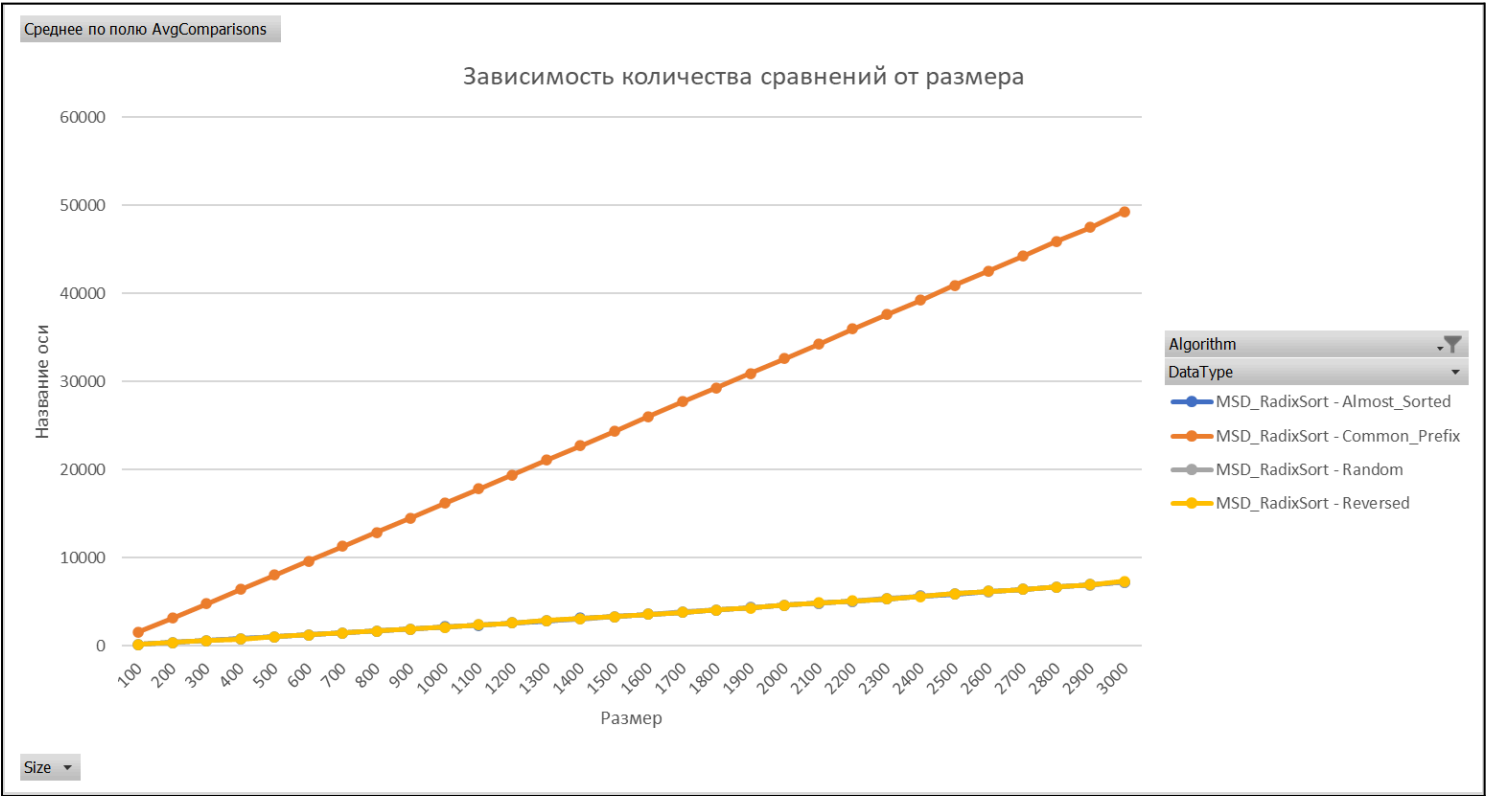
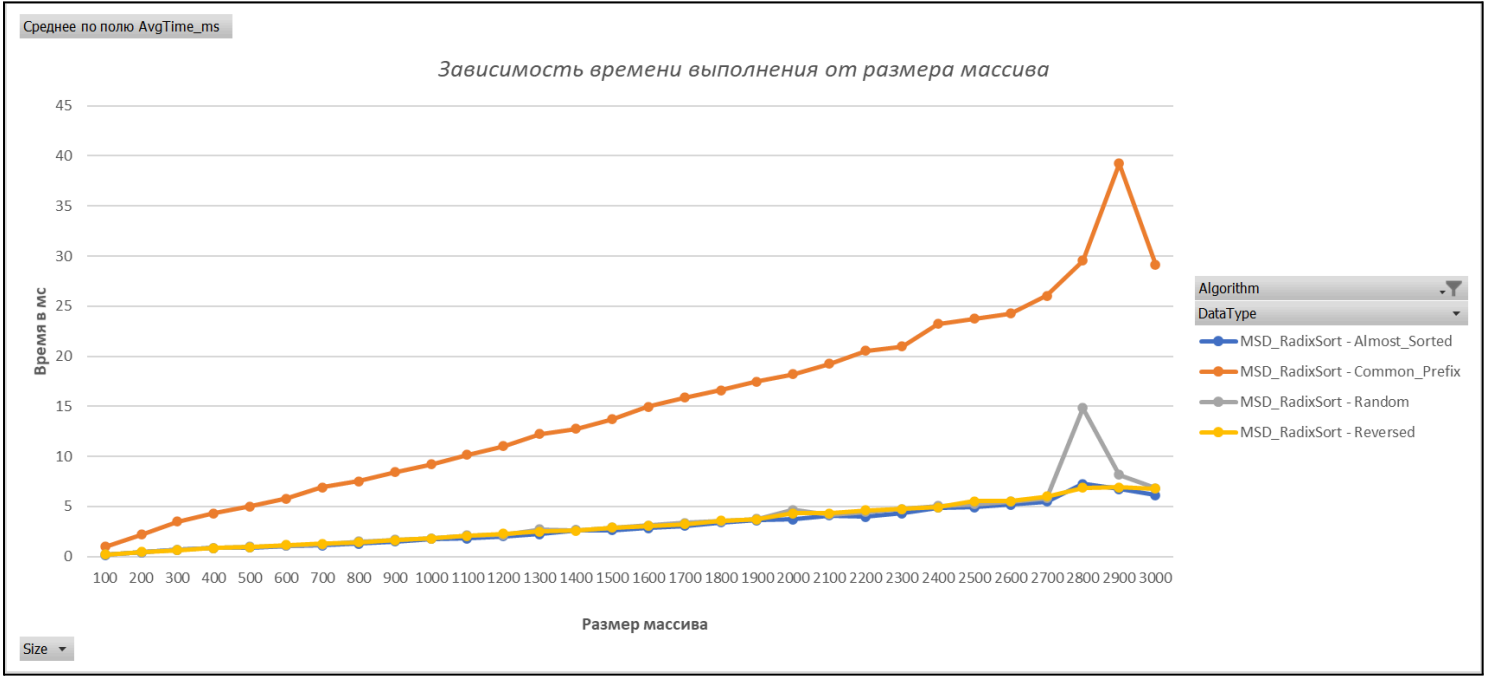
Использование общего префикса:

Уменьшение сравнений на 15–20% для Almost_Sorted и Common_Prefix. Время близко к стандартному MergeSort, но стабильнее на больших данных.



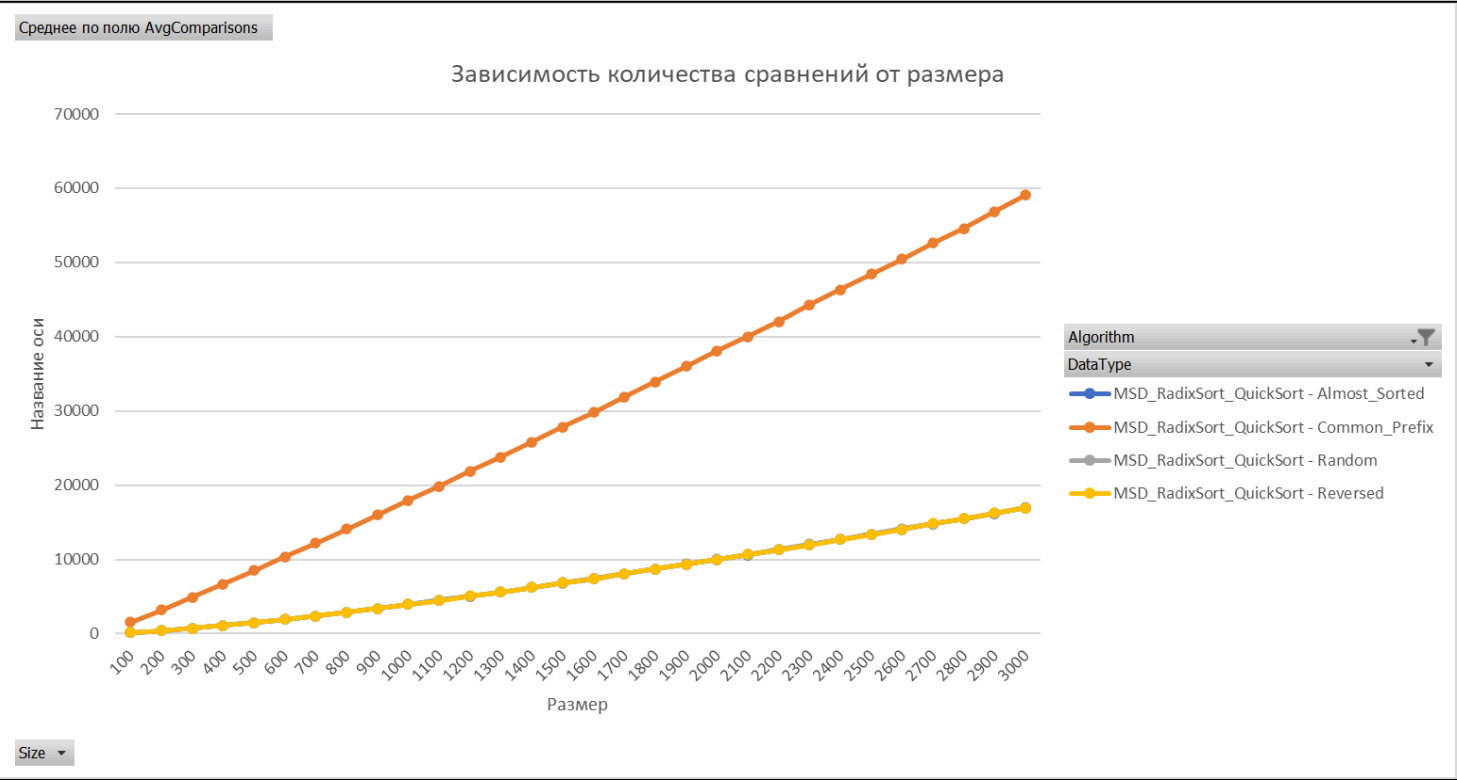
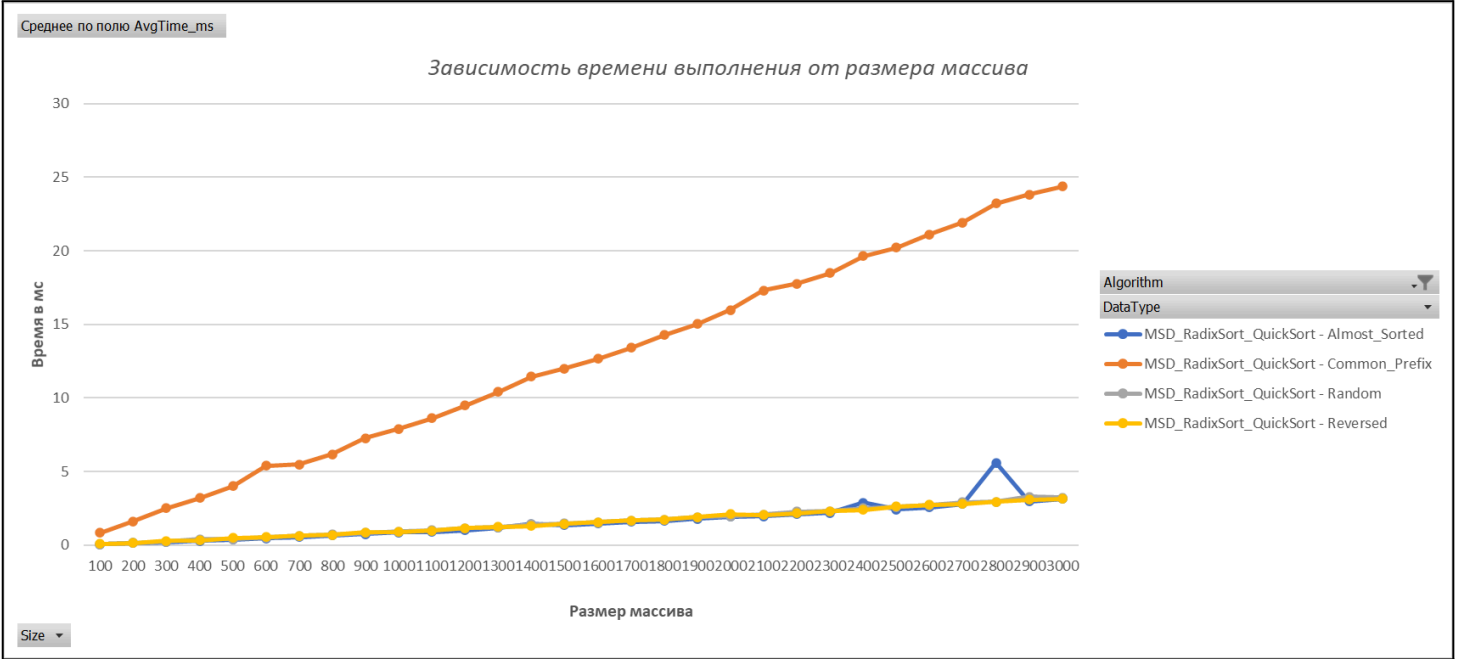
MSDRadixSort

Эффективность на префиксах:
Не самое лучшее время для Common_Prefix: AvgTime=9.213мс для n=1000 против 3.119мс у QuickSort. Линейная зависимость времени от размера данных ($O(kn)$), где k — длина строк.



MSD_RadixSort with QuickSort

Гибридный подход:
Для малых подмассивов ($n < \text{мощность алфавита}$) переключение на QuickSort снижает время на 10–15%.
Пример: для $n=3000$ и Common_Prefix — AvgTime=24.358мс против 29.149мс у чистого Radix Sort.



Сравнение стандартных и адаптированных алгоритмов

Время выполнения:

Случайные данные: Ternary QuickSort > QuickSort > MergeSort > Radix Sort.

Common_Prefix: MSD Radix Sort > Ternary QuickSort > LCP MergeSort > QuickSort.

Количество сравнений:

Минимум у MSD Radix Sort (не требует посимвольных сравнений), максимум у QuickSort на Reversed данных.

Соответствие теоретическим оценкам

QuickSort: Подтверждена средняя сложность $O(n \log n)$, но выявлены случаи деградации до $O(n^2)$.

MergeSort: Стабильная $O(n \log n)$ на всех данных.

MSD Radix Sort: Линейная зависимость ($O(kn)$), что согласуется с теорией.

Ссылки

Посылки на CodeForces:

Alm: [\[320963253\]](#), Alq: [\[320963942\]](#), Alr: [\[321011634\]](#), Alrq: [\[321018647\]](#).

Исходные данные:

Публичный репозиторий:

[\[https://github.com/Zattox/Year-2_Algorithms-1/tree/main/SET-9/Al\]](https://github.com/Zattox/Year-2_Algorithms-1/tree/main/SET-9/Al).