**Зубарева Наталия**

**БПИ195**

**Сравнение алгоритмов сортировок**

# Введение

В данном микроисследовании были проанализированы 8 сортировок на 4 типах массивов 10 возможных размеров. Было измерено среднее время работы (в наносекундах) каждого алгоритма на каждом виде массива каждого размера. Данные из основной таблицы (results.csv) сгруппированы в отдельные файлы – по одному на каждый алгоритм сортировки и на каждый тип данных в массиве. В соответствующих файлах также находятся графики, которые помогут нам сделать некоторые выводы.

Для начала рассмотрим данные и графики, относящиеся к каждому алгоритму сортировки, а затем проанализируем графики, сгруппированные по типу данных в массиве.

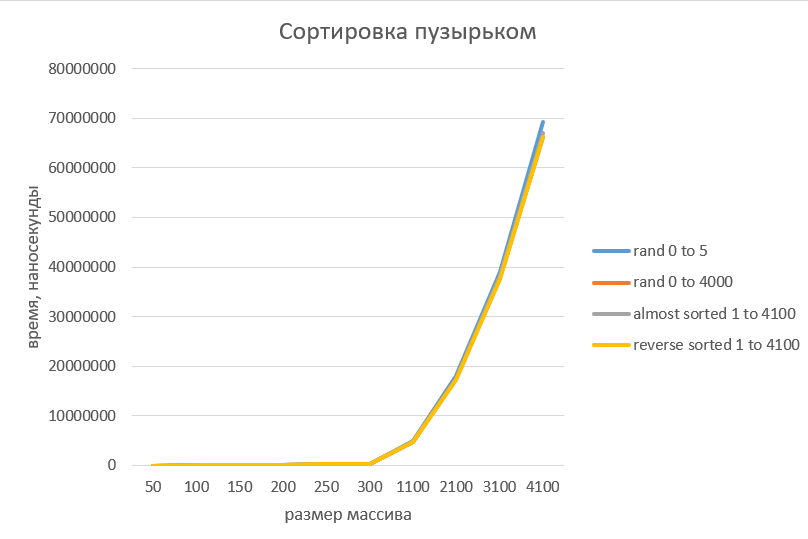
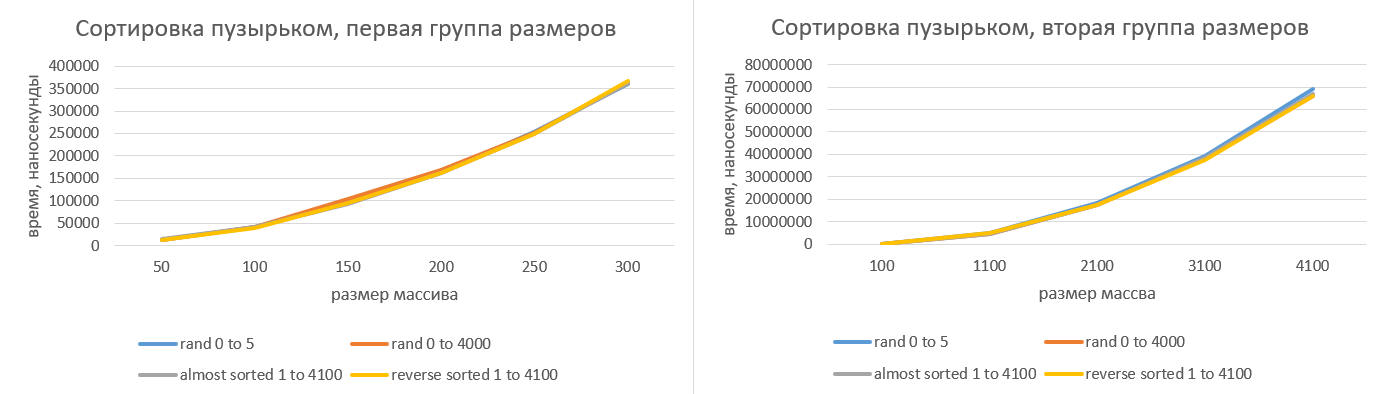
В блоках сортировки для каждой изображено 3 графика: один общий, отдельно на массивы размером от 100 до 300 с шагом 50, отдельно на массивы размером от 100 до 4100 с шагом в 1000. Разделение сделано с целью лучшего понимания скорости роста графика, потому что результаты на маленьких и больших размерах достаточно сильно отличаются, и их сложно воспринимать в одной и той же координатной сетке.

В блоках по видам данных графиков еще больше, а именно по 9 на каждый вид: один общий, один для всех алгоритмов сортировок на маленьких размерах массивов, один для всех алгоритмов на больших размерах массивов, и далее для больших и маленьких массивов по три графика – для быстрых сортировок, средних по скорости сортировок и медленных сортировок. Это сделано потому, что, как и в предыдущем пункте, смешивать разные размеры не очень репрезентативно, в то же время при отображении всех видов алгоритмов на одном графике они часто сливаются в неразличимые группы. Различия между быстрыми сортировками и медленными сортировками настолько велики, что сложно сравнивать сортировки внутри одной группы, для этого нужно построить на каждую отдельный график.

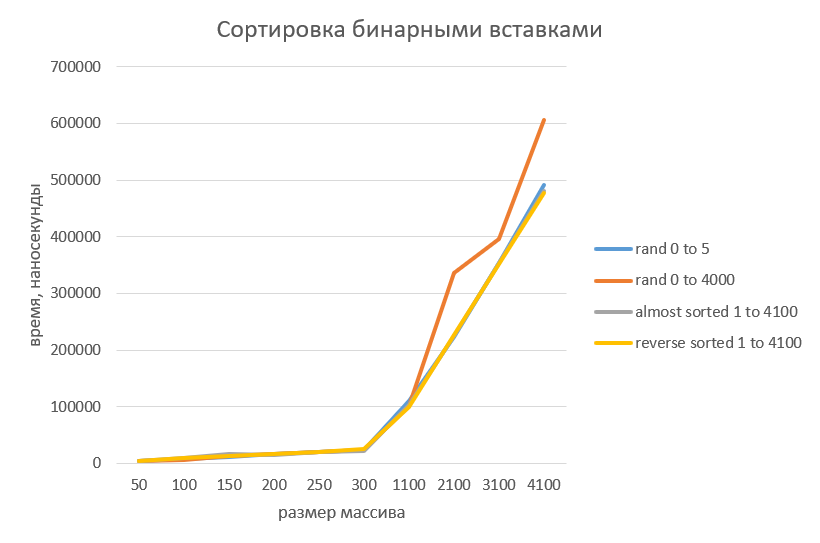
# Итерационные сортировки

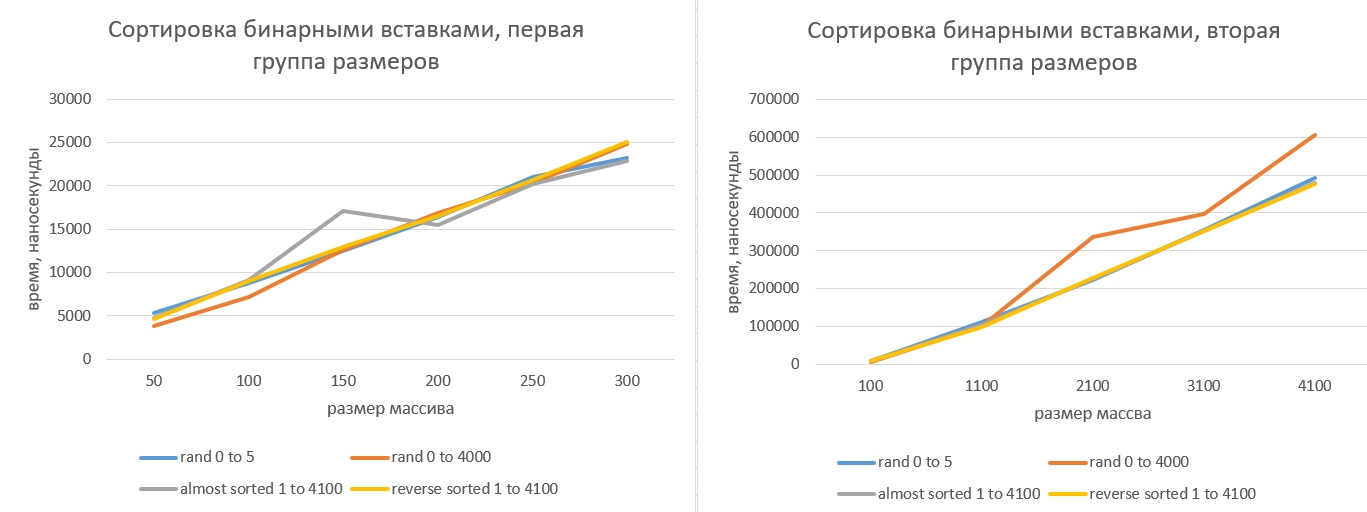
Сортировка пузырьком

- сортировка, с завидным постоянством одинаково медленно сортирующая все на свете. Такое ощущение, что разница в результатах в зависимости от размера и типа массива вообще не наблюдается, хотя на больших размерах массивы случайно сгенерированных чисел от 0 до 5 как будто сортируются в среднем дольше остальных, но разница пренебрежимо мала. Между самым маленьким и большим размером время увеличивается почти в 3000 раз.

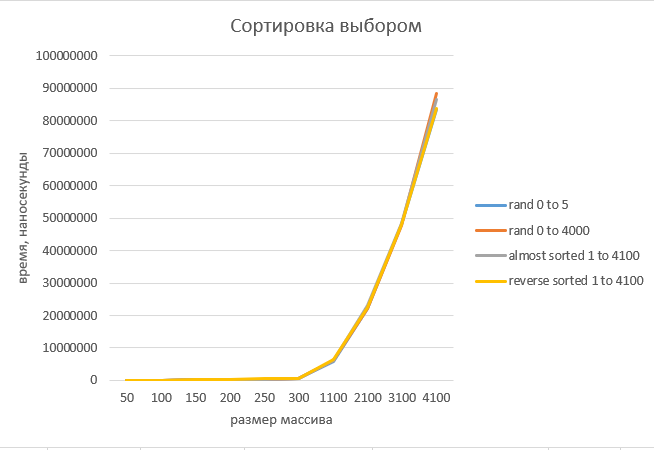
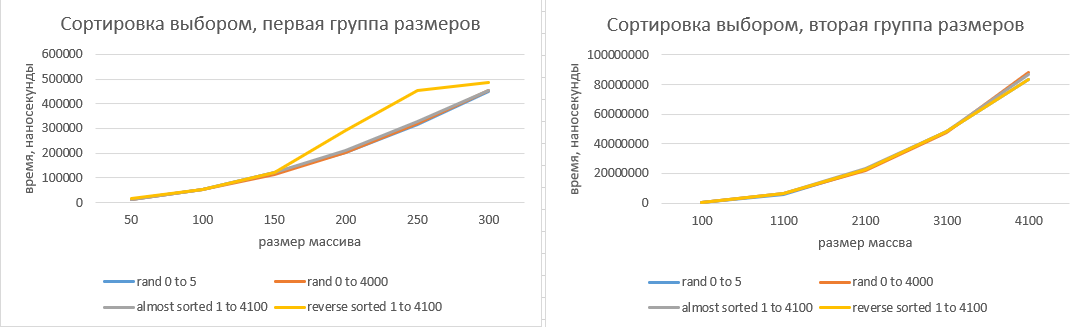
Сортировка бинарными вставками   
- тоже кажется сортирует все примерно одинаково, хотя массивы случайных чисел от 0 до 4000, сортируемые быстрее всего на самых маленьких размерах, занимают больше всего времени, начиная с размера в 1100 элементов. Все остальнеые типы массивов следуют примерно одному и тому же тренду, при этом между самыми маленькими размерами и самыми большими время увеличивается в 120 раз.





Сортировка выбором

- как и предыдущие две, очень непоколебимая сортировка. Все виды массивов сортируются примерно одинаково, за исключением обратно отсортированных массивов размером от 150 до 300 (что может быть какой-то статистической ошибкой, потому что объяснение этому придумать сложно). От наименьшего до наибольшего размера время сортировки растет в 9000 раз.

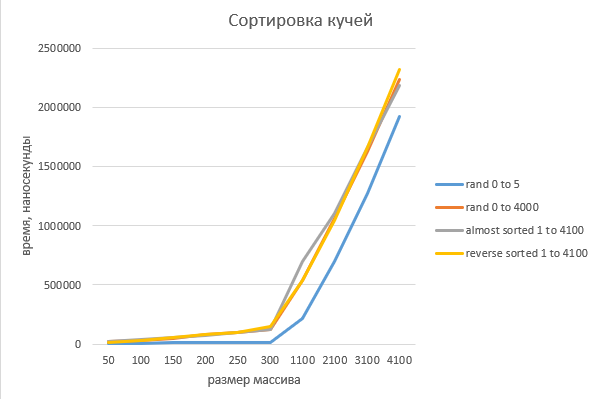
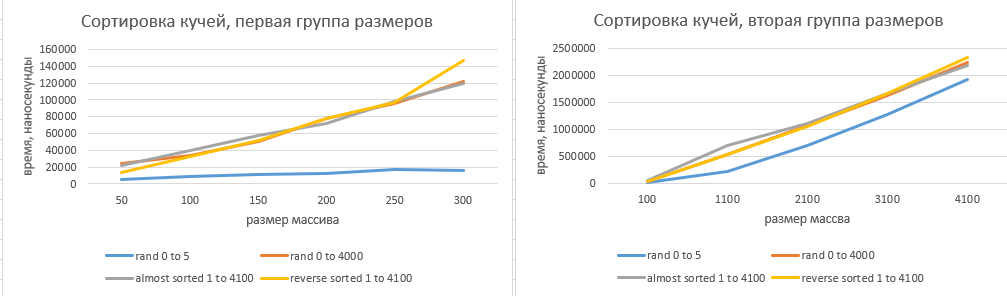
 

Можно предположить, что итерационные сортировки не очень подвержены влиянию типа данных, и что сортировка выбором сильнее всего замедляется при увеличении размера.

# Рекурсивные сортировки

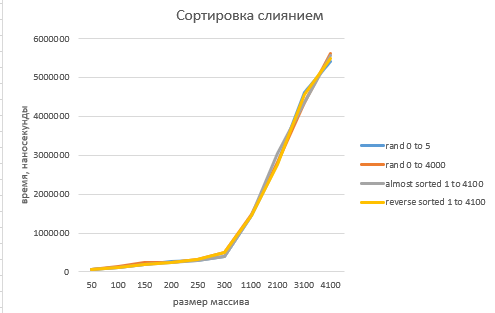
Сортировка кучей

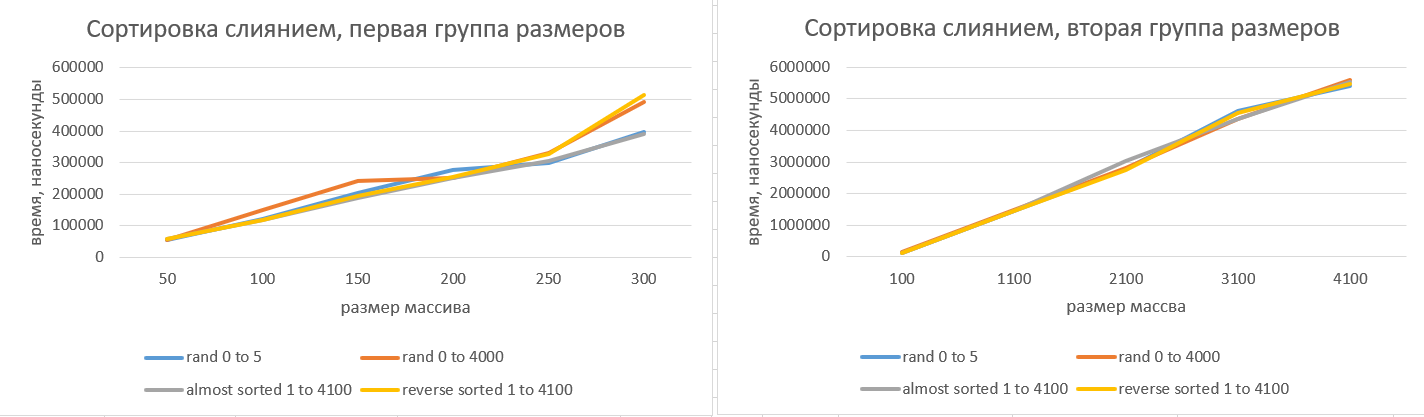
- явно быстрее всего сортирует массивы из случайных чисел от 0 до 5 при любом размере, при этом увеличение времени при увеличении размера вообще сравнительно медленное. Кажется, незначительно медленнее других сортируются обратно отсортированные массивы. В целом на худшем случае время сортировки от самого маленького до самого большого размера увеличивается в 250 раз.

Сортировка слиянием

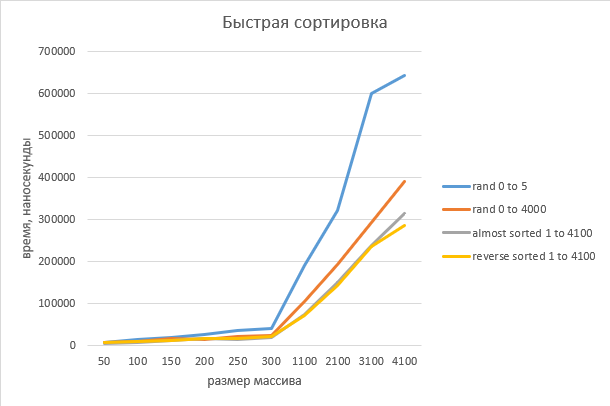
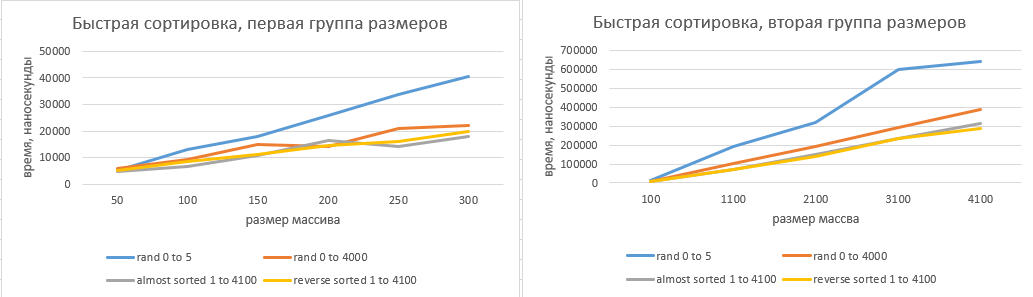
- почти не зависит от типа данных. На размерах от 200 до 300 почти отсортированные массивы как будто сортируются несколько лучше других, но в целом все линии очень похожи. Время растет примерно в 115 раз между самым маленьким и самым большим размером.





Быстрая сортировка

- действительно оказалась быстрой. Хуже всего сортирует массивы, состоящие из случайных чисел от 0 до 5, несколько быстрее – от 0 до 4000. Почти отсортированные массивы и обратно отсортированные обрабатываются одинаково быстро и быстрее остальных. В худшем случае время в зависимости от размера растет в 130 раз.

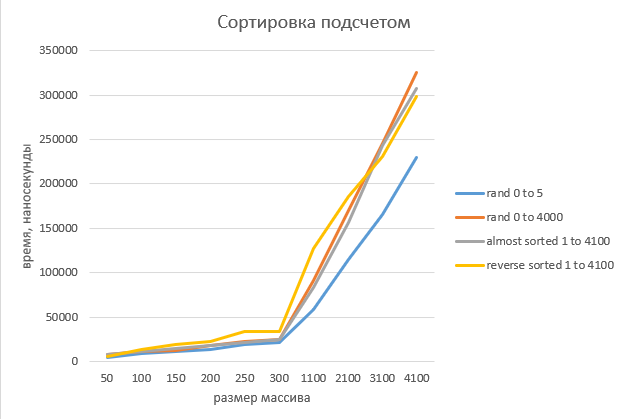
 

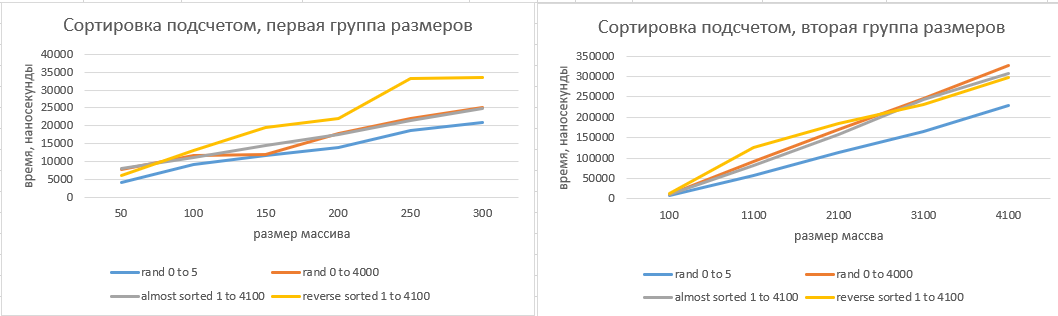
Таким образом, из линейных сортировок наиболее сильный рост в зависимости от времени у сортировки кучей, быстрее всего все же быстрая. Удивительно, но случайные числа от 0 до 5 одновременно – самые плохие для быстрой сортировки и самые лучшие для сортировки кучей.

# Линейные сортировки

Сортировка подсчетом

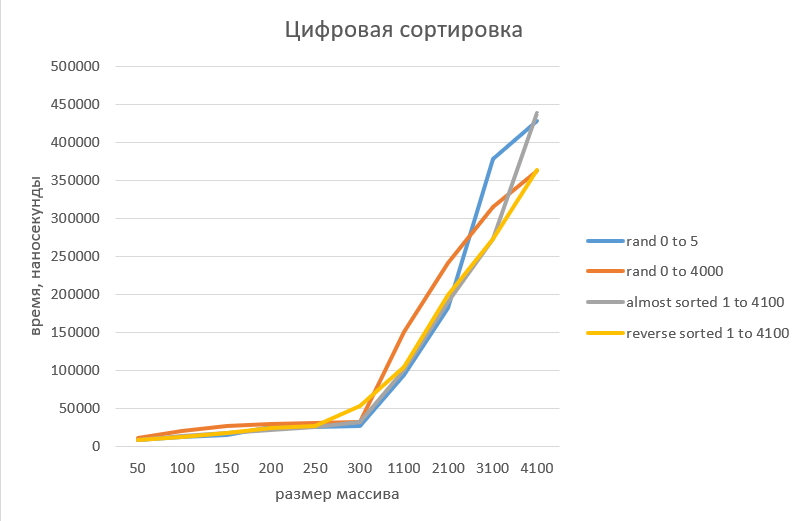
- быстрее всего сортирует массивы случайных чисел от 0 до 5 любых размеров, также при размере до 3100 медленнее всего сортирует обратно отсортированные массивы (но на больших размерах все типы кроме случайных от 0 до 5 примерно одинаковы). От самого маленького до самого большого размера время растет в 65 раз, вообще довольно быстрая.

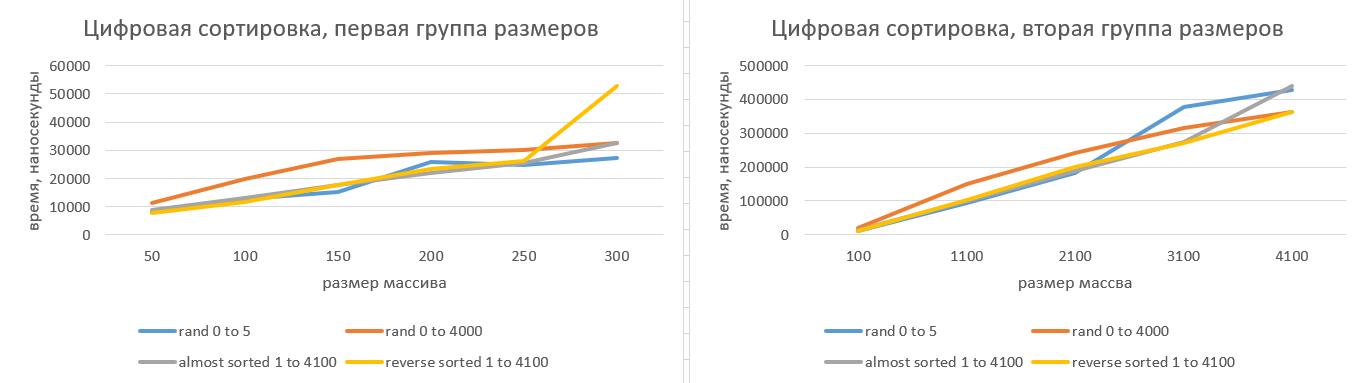




Цифровая сортировка

- удивительно хаотичная в плане предпочтений сортировка, как будто в целом медленнее сортирует массивы случайных чисел, но какую-то логику определить сложно. Время растет в 45 раз, но начинается с больших значений.





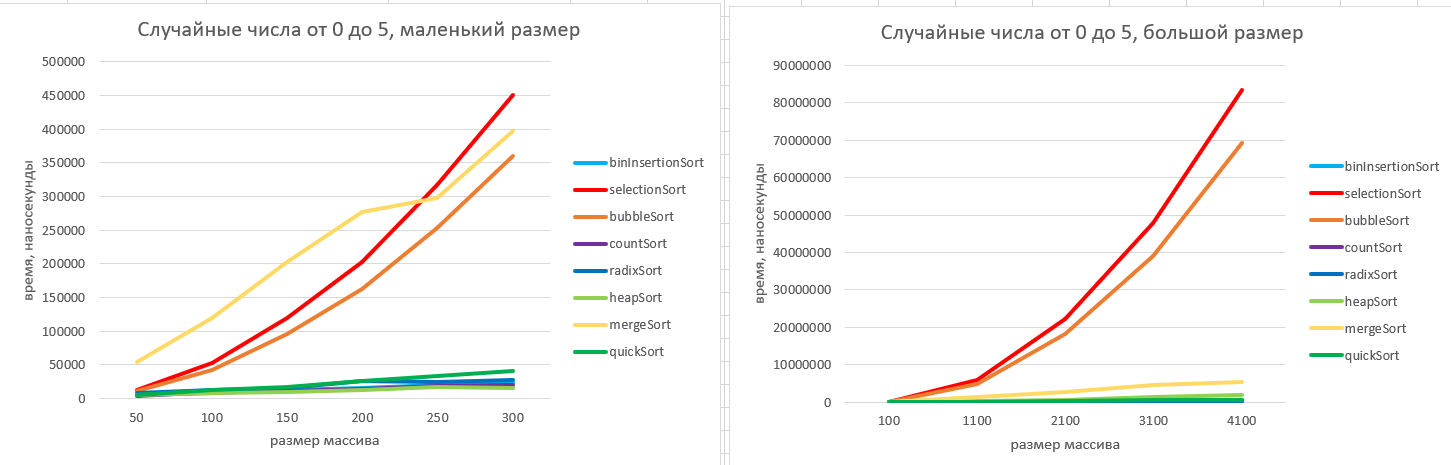
Кажется, из линейных сортировок цифровая хотя и растет по времени немного меньше, все же оказывается несколько медленнее сортировки подсчетом.

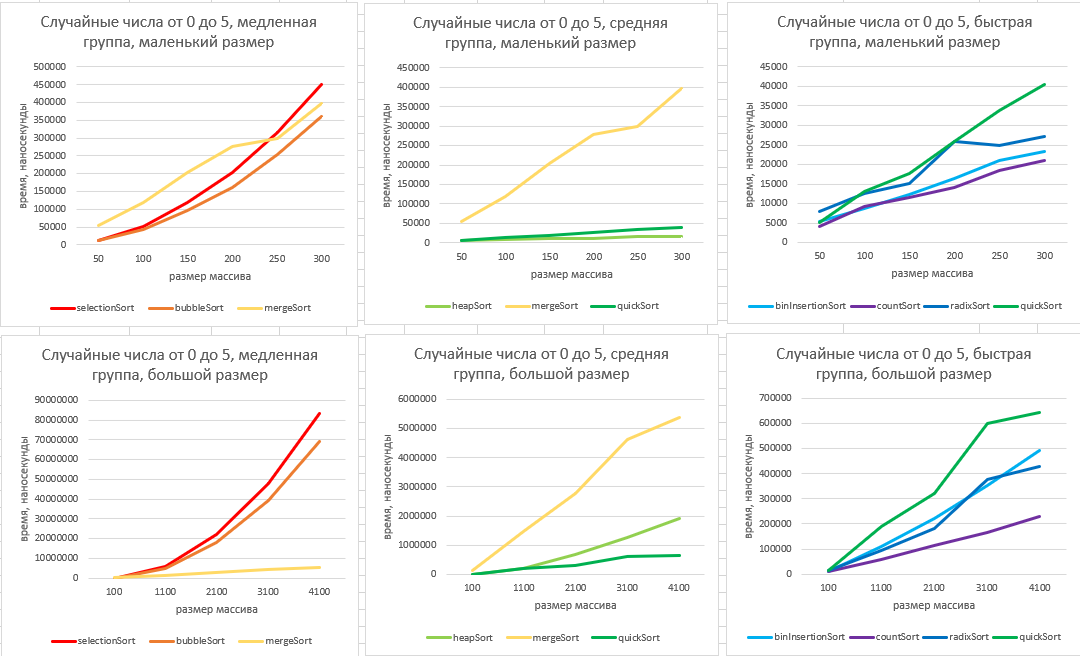
Из трех групп сортировок линейные меньше всего растут по времени в зависимости от размера, больше всего – итерационные (за исключением сортировки бинарными вставками, которая удивительно хороша для своей группы). Медленнее всего по максимальным значениям сортировка выбором, быстрее всех – сортировка подсчетом.

# Массив, заполненный случайными числами от 0 до 5

- явно хуже всего на нем сортировка выбором и сортировка пузырьком. На маленьких размерах также не очень хороша сортировка слиянием, но далее она реабилитируется. Быстрее всего и на маленьких, и на больших размерах – сортировка подсчетом, далее – цифровая и бинарными вставками, быстрая сортировка. Максимум времени – 85000000 нс, минимум – 5000.

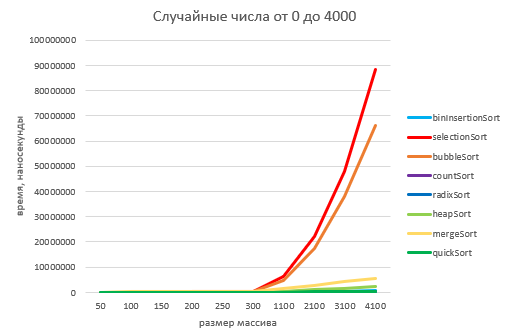


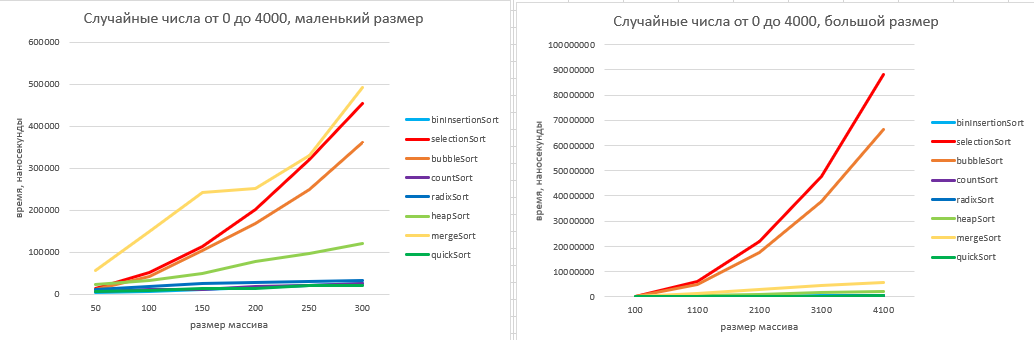




# Массив, заполненный случайными числами от 0 до 4000

- так же, как и прошлый, медленнее всего сортируется сортировкой выбором, пузырьком, хотя на маленьких размерах сортировка слиянием хуже. Хотя для маленьких размеров быстрее всего бинарные вставки и быстрая сортировка, на больших размерах сортировка подсчетом, быстрая сортировка и цифровая одинаково и наиболее быстрые. Также сортировка кучей, средняя на маленьких размерах, в числе средних на больших. Максимум времени – 90000000 нс, минимум – 5000.

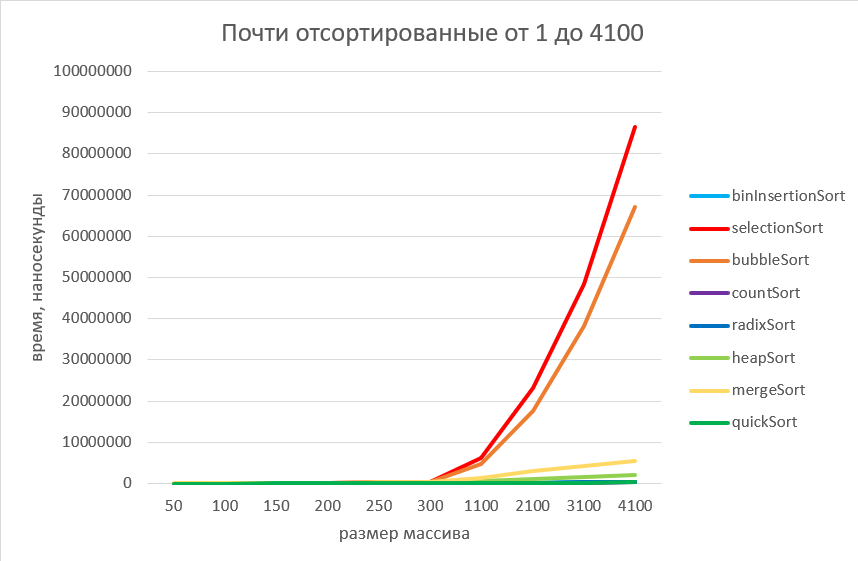


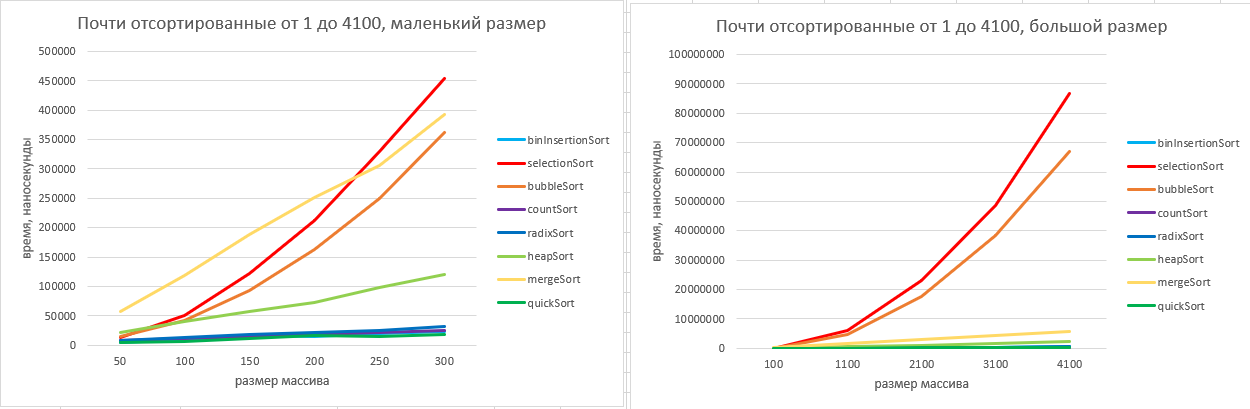


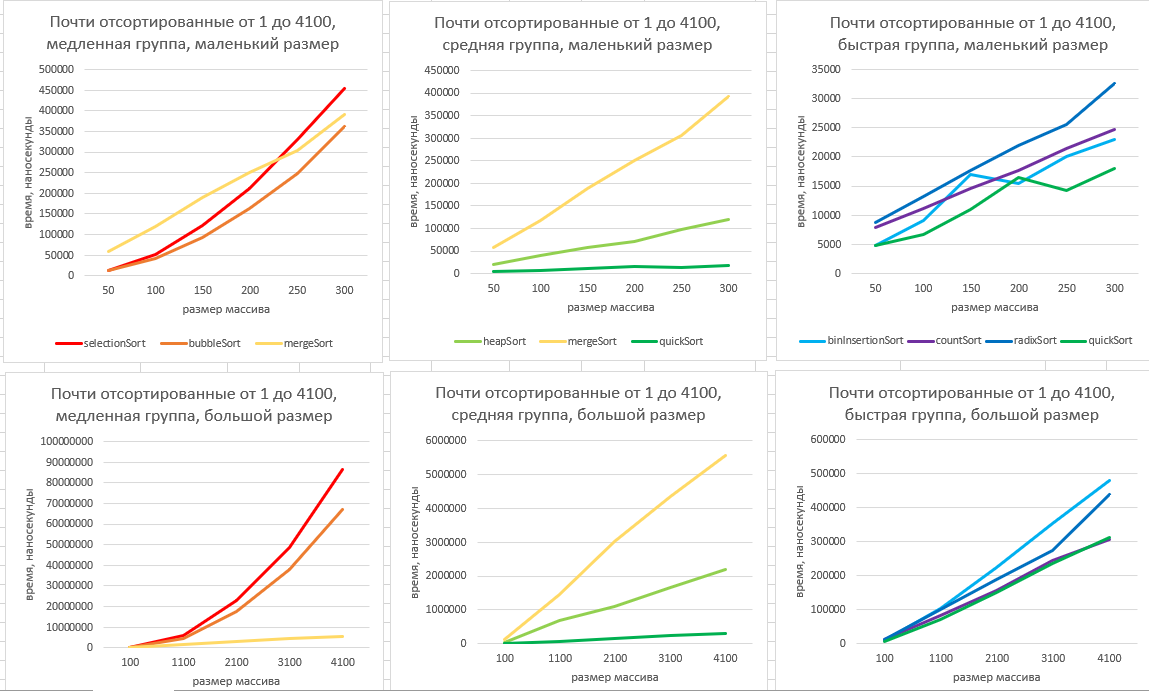


# Массив, заполненный числами от 1 до 4100, почти отсортированный

- все то же самое, что и у предыдущих, но на маленьких размерах явно лучше других быстрая сортировка, а на больших они сравниваются с сортировкой подсчетом. Максимум времени – 90000000 нс, минимум – 5000.

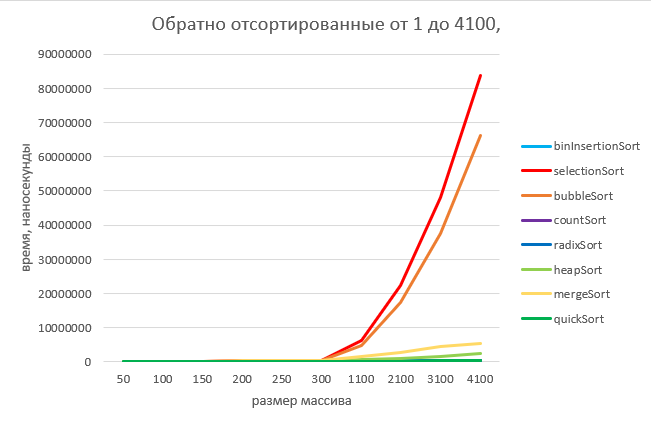


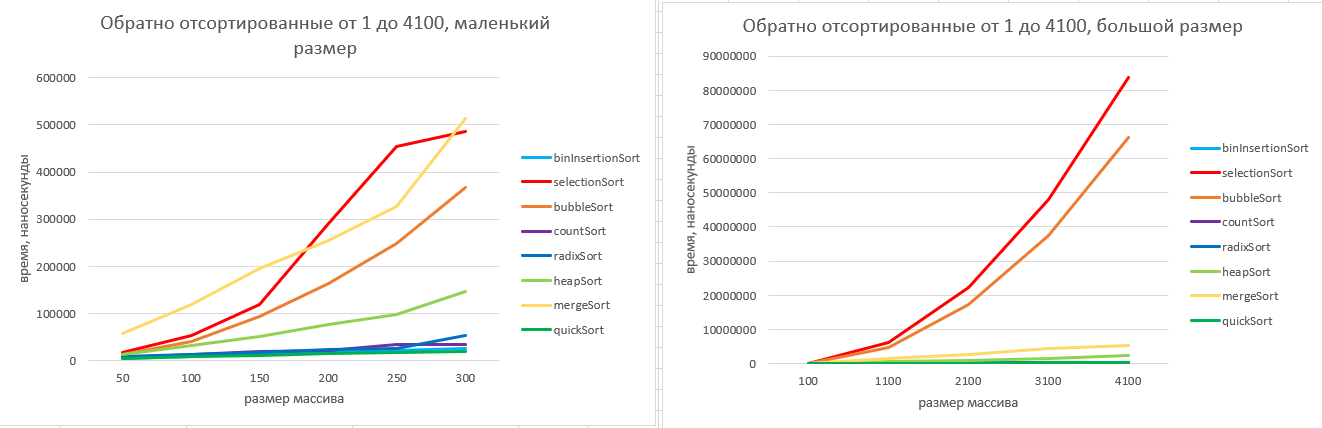


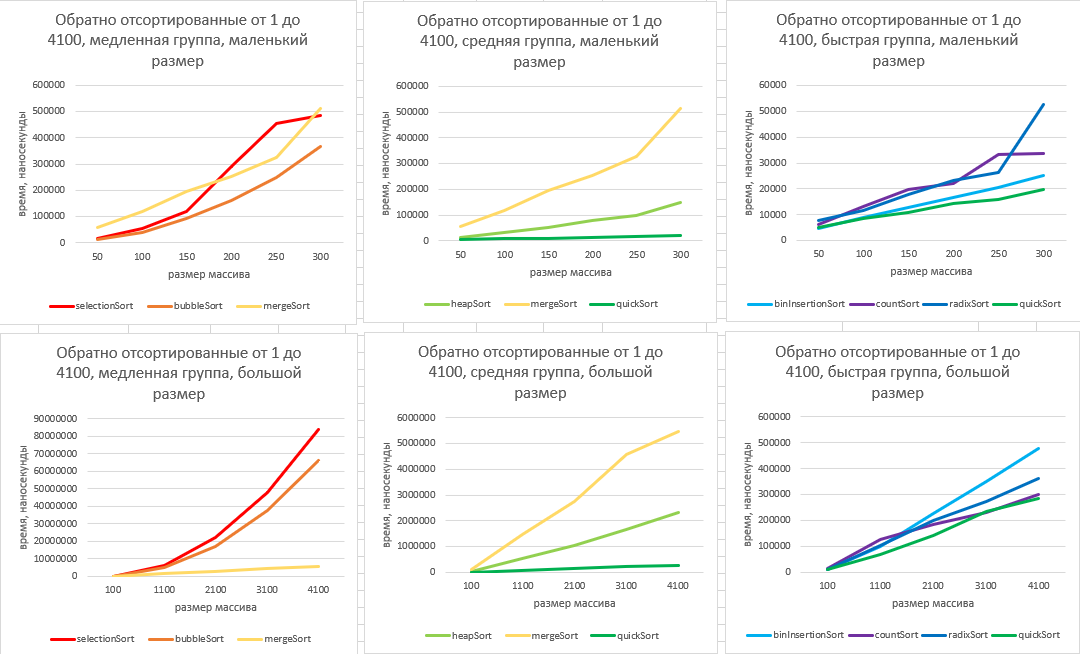


# Массив, заполненный числами от 1 до 4100, обратно отсортированный

- аналогично предыдущим, но на любых размерах сравнительно лучше других быстрая сортировка. Сортировки выбором и пузырьком – самые медленные, сортировка слиянием медленная на маленьких размерах, но достаточно неплоха на больших. Максимум времени – 90000000 нс, минимум – 5000.







В целом можно сделать несколько выводов: рекурсивные сортировки быстрее остальных, с ними наравне быстрая сортировка и сортировка бинарными вставками. Сортировка выбором медленная примерно на всех данных. На маленьких размерах еще не очень хороша сортировка слиянием, но на больших размерах она скорее средняя по скорости. Для всех размеров средняя сортировка кучей. В то время, как многим сортировкам вообще не важно, какие данные в массиве, наиболее часто крайними случаями были массивы из случайных чисел (медленный случай для сортировки бинарными вставками, цифровой сортировки, быстрой сортировки, быстрый случай для сортировки кучей, сортировки подсчетом). Обычно на время влияет скорее размер и алгоритм, чем тип данных.