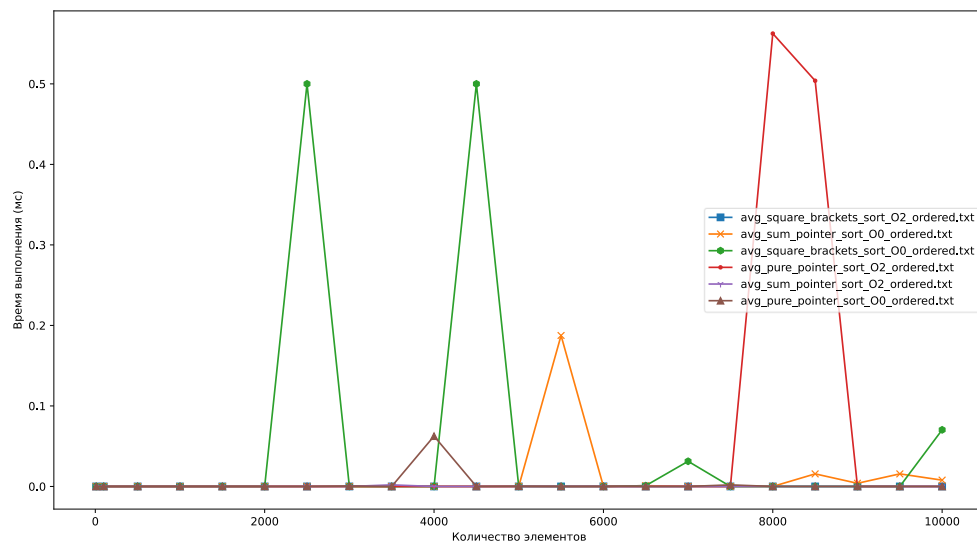


Постановка замерного эксперимента

Графики:

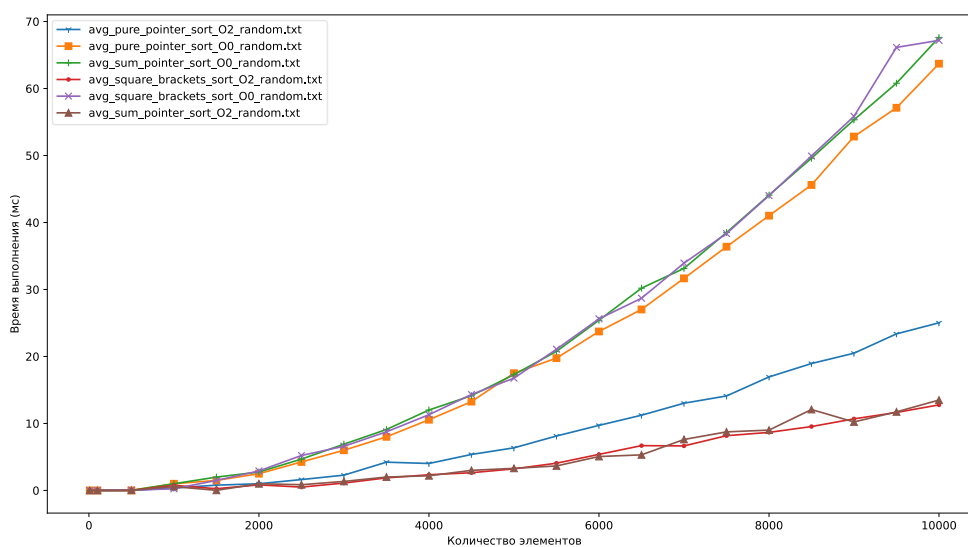
Среднее время сортировки массива в наилучшем случае (отсортированный массив)



N	*p O0	*p O2	a[] O0	a[] O2	*(p+n) O0	*(p+n) O2
10	0,0E+00	0,0E+00	0,0E+00	0,0E+00	0,0E+00	0,0E+00
100	0,0E+00	0,0E+00	0,0E+00	1,3E-51	5,6E-104	0,0E+00
500	1,0E-87	0,0E+00	1,3E-57	1,3E-23	3,4E-21	0,0E+00
1000	8,2E-143	4,0E-118	7,2E-71	7,0E-46	1,4E-42	1,1E-255
1500	4,4E-16	6,8E-21	7,5E-09	2,0E-118	5,0E-88	2,8E-135
2000	1,5E-39	3,2E-114	1,4E-20	2,2E-134	5,9E-54	8,5E-109
2500	7,9E-90	9,4E-38	5,0E-01	1,4E-247	4,0E-37	3,7E-96
3000	4,9E-04	1,9E-06	5,9E-39	5,2E-54	3,7E-40	2,8E-17
3500	1,9E-37	2,5E-57	4,0E-37	1,1E-22	6,0E-08	2,0E-03
4000	6,3E-02	9,3E-10	3,7E-09	2,5E-321	1,4E-20	4,1E-171
4500	8,7E-19	1,1E-19	5,0E-01	7,1E-161	7,8E-62	2,8E-76
5000	2,4E-04	7,7E-34	8,3E-25	1,3E-85	1,5E-11	7,2E-130
5500	1,1E-16	9,1E-13	2,1E-25	7,1E-74	1,9E-01	3,8E-37
6000	1,8E-15	1,4E-17	3,2E-30	1,2E-04	7,1E-15	3,6E-43
6500	1,5E-36	2,0E-15	9,8E-04	2,8E-45	9,8E-04	2,8E-73
7000	2,5E-32	8,5E-22	3,1E-02	8,3E-25	1,2E-04	1,5E-05
7500	2,0E-03	3,1E-89	4,7E-10	9,4E-38	1,3E-15	5,1E-57

8000	9,5E-07	5,6E-01	3,8E-37	1,9E-09	4,2E-22	7,3E-12
8500	2,2E-16	5,0E-01	7,6E-06	1,2E-60	1,6E-02	5,2E-113
9000	2,7E-20	4,7E-38	6,5E-55	1,7E-18	3,9E-03	1,7E-24
9500	3,6E-15	5,8E-11	7,7E-34	3,0E-08	1,6E-02	1,3E-29
10000	6,6E-24	4,2E-22	7,0E-02	1,5E-39	7,8E-03	2,4E-07

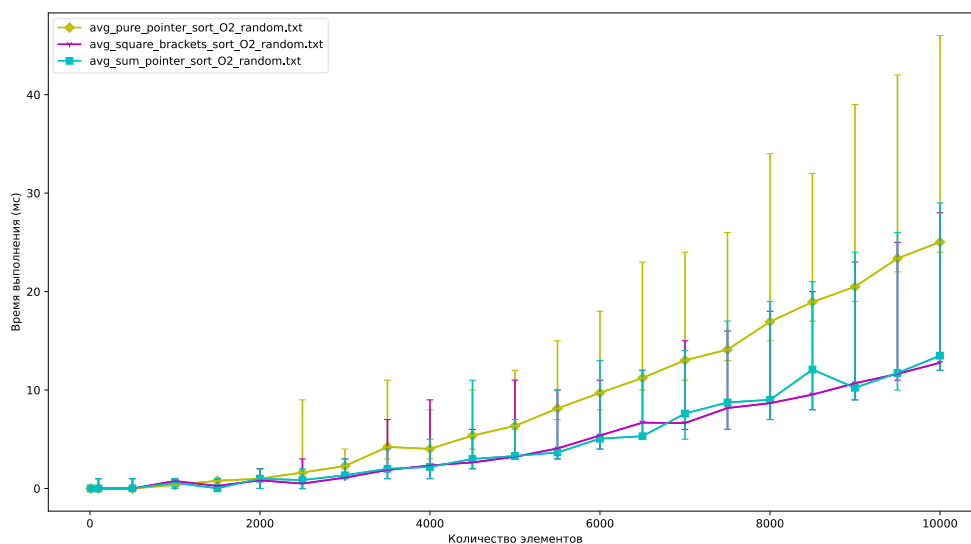
Среднее время сортировки случайно сгенерированных массивов



N	*p O0	*p O2	a[] O0	a[] O2	*(p+n) O0	*(p+n) O2
10	0,0E+00	0,0E+00	3,6E-304	0,0E+00	6,5E-173	0,0E+00
100	2,5E-116	5,3E-287	1,5E-126	2,9E-42	3,0E-08	2,1E-171
500	7,9E-03	1,2E-04	3,6E-02	1,9E-06	1,6E-02	1,6E-02
1000	9,9E-01	3,6E-01	2,5E-01	7,6E-01	1,0E+00	5,7E-01
1500	1,5E+00	7,9E-01	1,5E+00	2,6E-01	2,0E+00	1,8E-02
2000	2,5E+00	1,0E+00	2,9E+00	8,3E-01	2,7E+00	9,9E-01
2500	4,3E+00	1,6E+00	5,2E+00	5,0E-01	4,7E+00	8,6E-01
3000	6,0E+00	2,3E+00	6,6E+00	1,1E+00	6,9E+00	1,3E+00
3500	8,0E+00	4,2E+00	8,8E+00	1,9E+00	9,1E+00	2,0E+00
4000	1,1E+01	4,0E+00	1,1E+01	2,3E+00	1,2E+01	2,2E+00
4500	1,3E+01	5,4E+00	1,4E+01	2,6E+00	1,4E+01	3,0E+00
5000	1,8E+01	6,4E+00	1,7E+01	3,2E+00	1,7E+01	3,3E+00
5500	2,0E+01	8,1E+00	2,1E+01	4,1E+00	2,1E+01	3,7E+00
6000	2,4E+01	9,7E+00	2,6E+01	5,4E+00	2,5E+01	5,1E+00

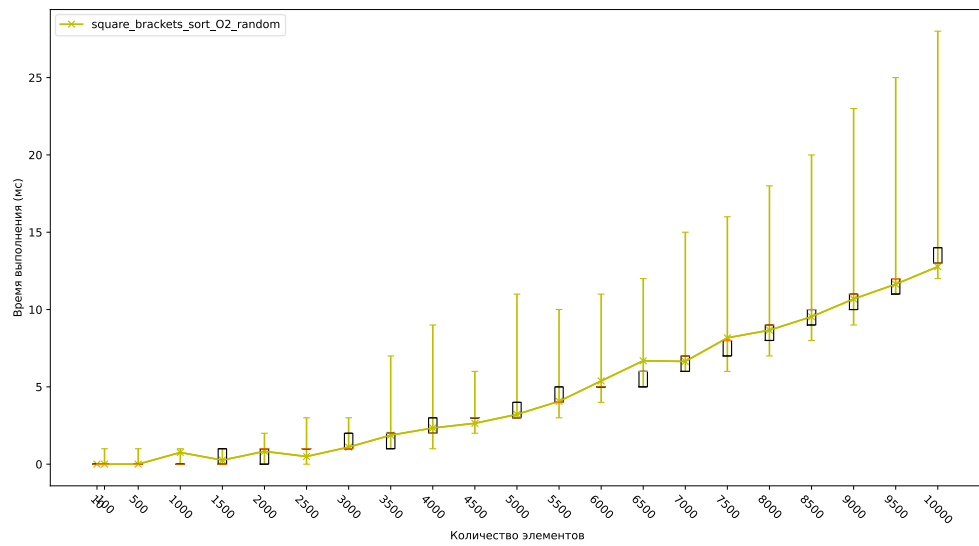
6500	2,7E+01	1,1E+01	2,9E+01	6,7E+00	3,0E+01	5,3E+00
7000	3,2E+01	1,3E+01	3,4E+01	6,6E+00	3,3E+01	7,6E+00
7500	3,6E+01	1,4E+01	3,8E+01	8,2E+00	3,8E+01	8,7E+00
8000	4,1E+01	1,7E+01	4,4E+01	8,7E+00	4,4E+01	9,0E+00
8500	4,6E+01	1,9E+01	5,0E+01	9,5E+00	5,0E+01	1,2E+01
9000	5,3E+01	2,0E+01	5,6E+01	1,1E+01	5,5E+01	1,0E+01
9500	5,7E+01	2,3E+01	6,6E+01	1,2E+01	6,1E+01	1,2E+01
10000	6,4E+01	2,5E+01	6,7E+01	1,3E+01	6,8E+01	1,3E+01

График «с ошибкой» для всех вариантов обработки массива при уровне оптимизации O2.



Средние значения указаны в таблице выше

График «с усами» с типом доступа квадратные скобки и уровнем оптимизации O2



n	tn	$\ln(t_{i+1}) - \ln(t_i) / \ln(n_{i+1}) - \ln(n_i)$
10	0	nan
100	2,9E-42	51,24162036
500	0,0000019	18,60964047
1000	0,76	-2,645447859
1500	0,26	4,034815447
2000	0,83	-2,271262599
2500	0,5	4,324542716
3000	1,1	3,54551603
3500	1,9	1,430788918
4000	2,3	1,040916643
4500	2,6	1,97075122
5000	3,2	2,600311577
5500	4,1	3,165241022
6000	5,4	2,694918489
6500	6,7	-0,202918484
7000	6,6	3,146187433
7500	8,2	0,917109198
8000	8,7	1,451040358
8500	9,5	2,564862535
9000	11	1,609318455
9500	12	1,560490677
10000	13	no data

Объяснение результатов

Всего для каждого варианта сортировки было проведено 1500 тестов (за 2 часа) с генерацией новых входных данных после каждого третьего замера

Среднее время сортировки отсортированных массивов варьируется на моей машине и с моим алгоритмом (сортировка вставками) в промежутке от 0 до 1 мс вне зависимости от оптимизации, причём возникновение единиц не поддаётся анализу (по крайней мере поверхностному), поэтому я спису их возникновение на случайность.

Среднее время сортировки случайных массивов

Без оптимизации:

В целом все варианты обращения к массиву занимают примерно одно и то же время, однако случай с использованием ссылок (и итерирования по ним) всё же стабильно выдавал более низкий результат. Это связано с тем, что машине не приходилось каждый раз считать сдвиг (или очередную ссылку) при обращении к очередному элементу массива.

С оптимизацией:

Картина меняется в строго противоположную сторону. Использование «чистых» указателей стабильно медленнее своих аналогов. Я полагаю, что это связано с тем, что оптимизатору проще оптимизировать случаи, где обращение с массивами ещё не приведено к ссылке. Это, разумеется, странно и у моих одноклассников такого эффекта не наблюдается, однако я перепроверил код и не смог найти серьёзной ошибки с моей стороны.

График с ошибкой

Средние числа (их анализ) были описаны мной в предыдущем пункте, про график с ошибкой (минимумы и максимумы) можно сказать лишь то, что среднее время как правило не сильно превышает минимальное (на единицы), однако максимальное время выполнения может подниматься гораздо выше, чем среднее число. Это можно списать на статистический выброс. Чтобы свести вероятность такой ошибки к минимуму, я проводил тестирование на ноутбуке и закрыл все программы кроме консоли, в которой непосредственно и проводилось тестирование. Ноутбук был переведён в режим производительности и на протяжении всего тестирования был подключен к источнику питания и не использовался для других целей.

График «с усами»

На графике с усами можно видеть как среднее с ошибкой, так и квартили, по которым можно понять примерное распределение полученных результатов. Оранжевыми линиями в прямоугольниках отмечена медиана и, так как мы исследуем квадратичную сортировку, если рассмотреть медианные значения, очертания параболы будут прорисовываться чётче, чем в случае графика по среднему. Также видно, что большая часть измерений попадала в относительно небольшой промежуток и большие значения максимумов являются исключениями, «выпавшими» не больше нескольких раз за 1500 измерений.

Ответы на вопросы

Какой способ обработки быстрее и почему?

Быстрее способ обработки с помощью ссылок, так как машине не нужно пересчитывать адреса элементов массива с помощью сдвигов (сложения) или индексов.

2. В датасете обнаружена серия экспериментов с одним результатом. Можно ли заменить её одним экспериментом?

Нет, нельзя, так как в таком случае мы не сможем достоверно узнать, какое значение измерений является средним, не сможем правильно определить квантили, а так же не сможем понять, что является статистическим выбросом, а что – нет.

3. Если заполнение случайными числами массива (или любая другая инициализация) присутствует в каждом эксперименте, то почему Вы замеряете время только у целевого алгоритма?

Так как заполнение числами – во-первых заполнение массива не имеет отношение к работе самого алгоритма, а во-вторых – потому что заполнение числами линейно, а сортировка массива – квадратична и замер времени заполнения массива был бы линейным сдвигом графика (или значений измерений) вверх на некоторую константу, что может помешать сделать правильные выводы.