

## Исследование характеристик программного обеспечения

### 1. Задание №1

#### Задача :

Используя функцию `nanosleep` и каждый из четырёх известных Вам способов замеров времени (`gettimeofday`, `clock_gettime`, `clock`, `__rdtsc`), исследовать среднее значение времени выполнения вызовов функции `nanosleep` для различных задержек (обязательные задержки: 1с, 100мс, 50мс, 10мс).

Для решения данной задачи необходимо реализовать следующие скрипты.

#### Обязательные скрипты:

- `build_apps.sh`, вызвав который, можно получить весь набор необходимых исполняемых файлов.
- `update_data.sh`, вызвав который, можно добавить некоторые данные в датасет данных исследования.
- `make_preproc.sh`, вызвав который, можно подготовить данные из набора, провести первичный анализ: посчитать среднее арифметическое, медианное, найти максимум и минимум, вычислить нижний и верхний квартили, etc.
- `go.sh`, вызвав который, можно получить данные исследования.

#### Дополнительные скрипты:

- `build_app.sh`, вызвав который, можно получить необходимые исполняемые файлы для одной конкретной программы на си.
- `update_specific_data_c.sh`, вызвав который, можно добавить дополнительные данные в датасет данных исследования для одной конкретной программы на си.

#### Таблица результатов исследования

Время задержки (мс)	Среднее время (мс)			
	<code>gettimeofday</code>	<code>clock_gettime</code>	<code>clock</code>	<code>__rdtsc</code>
1	1	1	573	2
5	5	5	689	10
10	10	10	751	21

15	15	15	714	31
20	20	20	804	42
30	30	30	750	63
50	50	50	778	104
70	70	70	837	146
100	100	100	852	209
300	300	300	825	628
500	500	500	909	1048
700	700	700	849	1467
800	800	800	814	1677
1000	1000	1000	792	2096

## 2. Задание №2

### Задача :

Проведите сравнение производительности работы программы по трём плоскостям:

- 1) Разные способы работы с элементами одномерного массива:
  - использование операции индексации  $a[i]$ ;
  - формальная замена операции индексации на выражение  $*(a + i)$ ;
  - использование указателей для работы с массивом.
- 2) Разные уровни оптимизации:
  - O0;
  - O2.
- 3) Разные исходные массивы:
  - наилучший случай (все элементы изначально отсортированы);
  - элементы расположены в случайном порядке.

Для решения данной задачи необходимо реализовать следующие скрипты.

### Обязательные скрипты:

- `build_apps.sh`, вызвав который, можно получить весь набор необходимых исполняемых файлов.
- `update_data.sh`, вызвав который, можно добавить некоторые данные в датасет данных исследования.
- `make_preproc.sh`, вызвав который, можно подготовить данные из набора, провести первичный анализ: посчитать среднее

арифметическое, медианное, найти максимум и минимум, вычислить нижний и верхний квартили, etc.

- `make_postproc.gpi`, вызвав который, можно получить указанные ниже графики.
- `go.sh`, вызвав который, можно получить данные исследования.

#### Дополнительные скрипты:

- `build_app.sh`, вызвав который, можно получить необходимые исполняемые файлы для одной конкретной программы на си.
- `generate_sort.sh`, вызвав который, можно получить текстовый файл, в котором содержится отсортированная по возрастанию последовательность чисел. Необходимое количество элементов последовательности задается пользователем.
- `generate_unsort.sh`, вызвав который, можно получить текстовый файл, в котором содержится неотсортированная последовательность чисел. Необходимое количество элементов последовательности задается пользователем.
- `update_specific_data_c.sh`, вызвав который, можно добавить дополнительные данные в датасет данных исследования для одной конкретной программы на си.
- `update_specific_data_exe.sh`, вызвав который, можно добавить дополнительные данные в датасет данных исследования для одного конкретного исполняемого файла.

#### Таблицы для наилучшего случая:

Для сортировки с индексами, оптимизация O0

Длина массива $n$	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0094	0	1
500	1	0,0170	1	5
1000	3	0,0526	2	47
1500	5	0,0673	3	56
2000	6	0,0503	5	45
2 500	8	0,1702	6	164

3 000	10	0,0608	7	37
3 500	12	0,0707	8	33
4 000	14	0,2221	10	193
4 500	15	0,1763	12	171
5 000	17	0,0970	12	44
5 500	20	1,0723	13	989
6 000	20	0,1630	15	90
6500	22	0,6711	15	677
7000	24	0,1210	18	62
7500	24	0,1451	18	59
8000	26	0,1260	20	56
8500	30	0,2873	22	223
9000	32	0,1991	24	84
9500	37	3,4983	25	3527
10000	37	0,7037	26	663
11000	39	0,5668	26	568
12000	43	0,2418	31	150
13000	45	0,2239	33	128
14000	49	0,2467	33	156
15000	52	0,2537	35	117
16000	55	0,2489	37	109
17000	59	0,3612	40	217
18000	60	0,3097	42	135
19000	59	0,3097	44	207
20000	57	0,2692	47	138

Для сортировки с индексами, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0090	0	1
500	0	0,0150	0	3
1000	1	0,0203	0	11
1500	2	0,0311	1	16
2000	2	0,0360	1	14
2 500	4	0,5322	2	529
3 000	4	0,4738	2	476
3 500	5	0,1900	3	182

4 000	6	0,5128	3	515
4 500	6	0,0673	3	16
5 000	7	0,1021	4	76
5 500	8	0,0846	4	23
6 000	8	0,0906	5	32
6500	9	0,1004	5	38
7000	9	0,1087	5	30
7500	10	0,1516	5	107
8000	11	0,1390	6	61
8500	12	0,2004	7	157
9000	14	0,3941	7	375
9500	14	0,1544	7	41
10000	15	0,1606	7	51
11000	16	0,1906	8	86
12000	18	0,2072	9	89
13000	19	0,3102	9	207
14000	20	0,2260	10	86
15000	21	0,2657	11	153
16000	22	0,2380	12	43
17000	25	0,2582	12	94
18000	24	0,3024	13	171
19000	23	0,2449	14	68
20000	24	0,2474	14	58

Для сортировки с выражениями, оптимизация ОО

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0094	0	1
500	1	0,0208	1	11
1000	3	0,0244	2	14
1500	5	0,0322	3	15
2000	6	0,0768	5	59
2 500	8	0,0587	6	47
3 000	10	0,2637	7	209
3 500	12	0,6305	9	31
4 000	14	0,6305	10	624
4 500	15	0,1140	12	96

5 000	17	0,6021	13	117
5 500	19	0,6021	14	612
6 000	20	0,5191	14	523
6500	22	0,1187	17	57
7000	23	0,1393	18	65
7500	25	0,4950	19	484
8000	26	0,1501	20	64
8500	29	0,1739	21	71
9000	33	0,2725	24	158
9500	34	0,2123	26	133
10000	36	0,1922	26	90
11000	39	0,1896	29	101
12000	44	0,7422	31	748
13000	47	0,6227	34	586
14000	49	0,2812	37	214
15000	52	0,2498	35	136
16000	57	1,8520	43	1871
17000	59	0,2575	44	123
18000	63	0,3592	42	250
19000	61	0,2870	44	156
20000	59	0,2366	47	97

Для сортировки с выражениями, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего.	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0080	0	1
500	0	0,0181	0	12
1000	1	0,0212	0	14
1500	2	0,0332	1	18
2000	3	0,1028	1	98
2 500	3	0,0823	2	76
3 000	4	0,0456	2	12
3 500	5	0,0882	3	70
4 000	5	0,0603	3	16
4 500	6	0,0898	3	63
5 000	7	0,0952	4	67
5 500	8	0,1929	4	161

6 000	8	0,1152	5	77
6500	9	0,1032	5	52
7000	10	0,0707	5	1028
7500	12	1,9899	5	1995
8000	11	0,4232	6	403
8500	12	0,1614	6	87
9000	14	0,1389	6	42
9500	14	0,1503	7	40
10000	15	0,1575	7	44
11000	17	0,2417	8	129
12000	24	6,9507	9	6966
13000	21	1,3039	9	1303
14000	20	0,2163	10	47
15000	21	0,2297	11	63
16000	23	0,2588	12	84
17000	24	0,2565	12	67
18000	24	0,2784	13	101
19000	24	0,2482	14	55
20000	25	0,4106	14	319

Для сортировки с указателями, оптимизация ОО

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0100	0	1
500	1	0,0381	1	36
1000	2	0,0270	2	14
1500	5	1,0251	3	1029
2000	5	0,0313	4	14
2 500	7	0,2625	5	265
3 000	8	0,1062	6	78
3 500	10	0,0858	7	60
4 000	11	0,0617	8	28
4 500	12	0,0928	10	78
5 000	14	0,0959	10	77
5 500	15	0,0832	12	50
6 000	17	0,1003	13	54
6500	18	0,1116	14	49

7000	19	0,1169	15	53
7500	20	0,1135	15	56
8000	22	0,1381	17	74
8500	25	0,1726	19	127
9000	26	0,1488	20	73
9500	28	0,1542	20	69
10000	29	0,1823	22	122
11000	32	0,1827	22	108
12000	38	3,6354	23	3648
13000	38	0,2059	29	120
14000	40	0,1826	28	92
15000	49	5,1284	29	5166
16000	47	0,2292	31	124
17000	48	0,4613	33	443
18000	49	0,2467	35	157
19000	47	0,2220	37	113
20000	48	0,2051	39	125

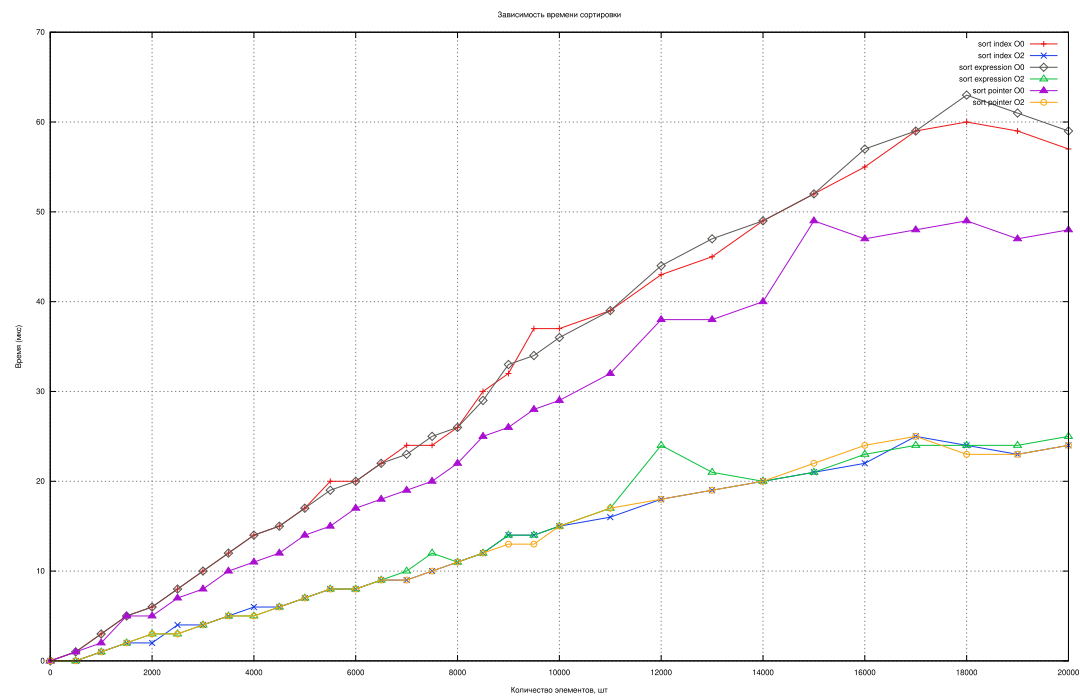
Для сортировки с указателями, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0087	0	1
500	0	0,0141	0	2
1000	1	0,0204	0	12
1500	2	0,0298	1	16
2000	3	0,6588	1	661
2 500	3	0,0705	2	55
3 000	4	0,0804	2	63
3 500	5	0,0822	3	59
4 000	5	0,0753	3	80
4 500	6	0,0960	3	35
5 000	7	0,0777	4	24
5 500	8	0,0835	4	31
6 000	8	0,0909	5	26
6500	9	0,1082	5	59
7000	9	0,1031	5	29
7500	10	0,1123	6	47



8000	11	0,1227	6	34
8500	12	0,1426	7	58
9000	13	0,1791	7	119
9500	13	0,1488	7	38
10000	15	0,1911	7	100
11000	17	0,1738	8	57
12000	18	0,2099	9	58
13000	19	0,1938	9	91
14000	20	0,2131	10	61
15000	22	0,5443	11	517
16000	24	0,6048	11	422
17000	25	0,2552	12	74
18000	23	0,2461	13	48
19000	23	0,2490	14	99
20000	24	0,7918	14	777

График обработки наилучшего случая



## Таблицы для общего случая:

Для сортировки с индексами, оптимизация ОО

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0097	0	1
500	164	3,0296	119	2909
1000	672	4,4071	495	2799
1500	1586	8,3454	1176	4380
2000	2652	10,0146	2070	4999
2 500	4340	15,5539	3287	6603
3 000	6466	21,0320	4778	13126
3 500	8488	26,2558	6410	12132
4 000	11337	31,4374	8598	14902
4 500	14367	42,0915	10625	22524
5 000	17841	44,8960	12834	26279
5 500	21077	57,6121	16547	27657
6 000	23595	55,8164	18966	32951
6500	28160	64,2125	22715	37671
7000	32317	68,8059	26418	42684
7500	35750	68,9295	29792	45047
8000	39498	65,5062	34887	49797
8500	43041	69,1447	37750	56009
9000	47233	66,1037	41649	55846
9500	52026	81,5265	46364	67154
10000	56885	83,1104	50907	72658
11000	67481	105,9312	59603	89587
12000	77588	109,0512	70509	99214
13000	89498	118,0596	82000	123182
14000	103254	139,2012	95458	131967
15000	116593	152,2047	108993	146422
16000	131589	145,4694	122825	167040
17000	146232	178,0790	138148	189939
18000	163941	224,2253	154658	209779
19000	180058	220,5157	170025	231819
20000	200245	276,1478	189590	259060

Для сортировки с индексами, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0089	0	1
500	76	1,9475	50	1886
1000	306	5,2314	205	4656
1500	690	6,1410	464	3136
2000	1170	9,3218	770	5776
2 500	1916	11,2100	1265	5015
3 000	2755	17,9649	1878	12636
3 500	3561	18,1453	2507	6564
4 000	4897	22,3842	3326	11877
4 500	5834	27,1749	4262	9154
5 000	7652	30,1254	5240	12541
5 500	9326	33,0855	6385	12971
6 000	11008	34,6069	7525	14000
6500	13111	40,6016	8697	18190
7000	14885	50,9691	10110	19278
7500	16181	63,1624	11487	28356
8000	19201	57,8832	13326	26839
8500	20186	55,6043	15620	25036
9000	22655	54,7397	18226	29254
9500	25129	51,3354	19261	33995
10000	27968	52,6284	22374	36422
11000	32623	72,9518	24613	42094
12000	36687	71,3389	30600	44229
13000	41190	60,4214	36973	52874
14000	46243	51,9256	42335	59689
15000	51265	54,0643	47385	68395
16000	56270	69,4363	52194	80367
17000	62240	67,8244	57928	88429
18000	67786	63,4774	64363	91808
19000	74905	59,7477	70907	86695
20000	81325	60,5163	78424	99713

Для сортировки с выражениями, оптимизация ОО

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0089	0	1
500	167	4,1012	122	3099
1000	665	4,4528	490	3055
1500	1563	7,4440	1185	4148
2000	2663	14,0247	1946	11939
2 500	4350	15,8226	3343	6850
3 000	6230	21,1786	4704	9028
3 500	8748	25,0663	6339	13273
4 000	10961	33,1591	8515	15299
4 500	14247	41,9699	10858	19107
5 000	18172	43,5300	13750	23305
5 500	20907	57,1605	16363	28404
6 000	23682	56,7838	19224	32154
6500	28224	63,8183	22975	36263
7000	32323	70,0082	26195	39988
7500	35790	72,7273	29492	46794
8000	39636	70,4200	33960	52537
8500	43255	64,6889	38203	53998
9000	47296	69,3940	42213	61811
9500	52194	79,3957	46218	70143
10000	56519	92,6711	50516	75940
11000	66583	104,7103	59728	88606
12000	77984	106,9224	70657	103158
13000	89415	145,5350	81901	120968
14000	103747	144,3362	95739	136964
15000	117467	172,0686	108824	153137
16000	130663	188,4603	122444	175159
17000	146424	181,2660	138368	186919
18000	163016	219,5894	154483	210109
19000	179703	233,7838	170521	233281
20000	200737	269,2831	189520	279169

Для сортировки с выражениями, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0092	0	1
500	71	0,9038	51	771
1000	295	2,4226	195	1385
1500	687	6,7994	449	5505
2000	1165	9,2449	785	4301
2 500	1884	10,2262	1339	3968
3 000	2695	13,9757	1880	7466
3 500	3643	17,3167	2590	6088
4 000	4805	23,5586	3227	12726
4 500	6092	31,3158	4142	13533
5 000	7625	28,2976	5199	10926
5 500	9123	35,9137	6291	15789
6 000	10950	37,8211	7604	15211
6500	12810	45,3441	8941	17732
7000	15000	49,4369	10336	19887
7500	16753	58,1301	11159	22577
8000	19093	59,4177	13401	27191
8500	19687	57,4396	15529	29396
9000	22688	54,3367	18623	29902
9500	25086	50,4315	20095	32203
10000	27875	49,8034	22190	39591
11000	32659	75,5038	23612	45313
12000	36765	70,2692	30035	48474
13000	41203	60,4257	34596	55657
14000	46184	58,6180	42390	62760
15000	51350	58,2998	47583	71250
16000	56370	60,9290	52078	74315
17000	62031	68,7945	58358	86913
18000	68048	64,0128	64388	99991
19000	74514	59,6829	67029	88034
20000	82534	60,4974	79133	100062

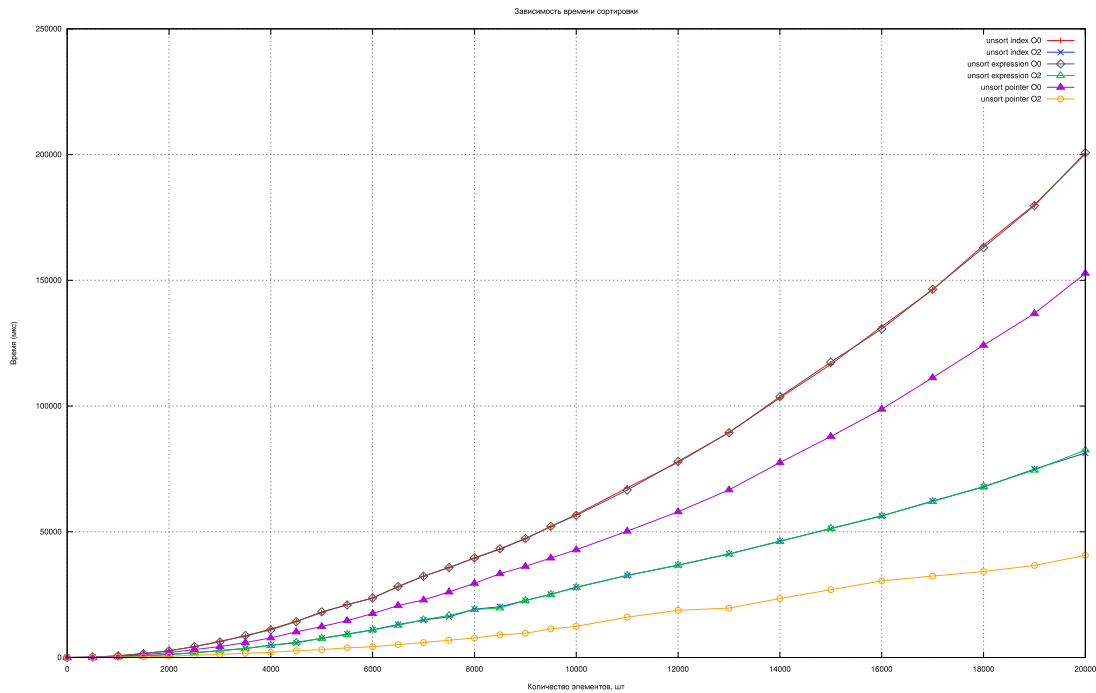
Для сортировки с указателями, оптимизация ОО

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0097	0	1
500	119	3,0296	96	1005
1000	482	4,4071	392	2461
1500	1119	8,3454	917	4759
2000	1914	10,0146	1404	7868
2 500	3153	15,5539	2453	15454
3 000	4341	21,0320	3513	6998
3 500	5921	26,2558	4613	9107
4 000	7799	31,4374	6258	10907
4 500	10178	42,0915	8448	14791
5 000	12308	44,8960	10027	16197
5 500	14671	57,6121	11909	20976
6 000	17468	55,8164	14446	23896
6500	20650	64,2125	16955	28505
7000	22876	68,8059	19071	30226
7500	26048	68,9295	22112	34708
8000	29499	65,5062	25713	42140
8500	33309	69,1447	27488	47966
9000	36268	66,1037	31498	52998
9500	39556	81,5265	35038	53909
10000	42828	83,1104	38708	56436
11000	50249	105,9312	45547	69246
12000	57953	118,0596	53267	85046
13000	66651	139,2012	62303	100501
14000	77527	152,2047	72583	113448
15000	87908	145,4694	82803	130030
16000	98765	165,6467	93493	146925
17000	111281	178,0790	105314	166915
18000	124172	224,2253	117964	190217
19000	136810	220,5157	130172	209700
20000	152821	276,1478	144934	231602

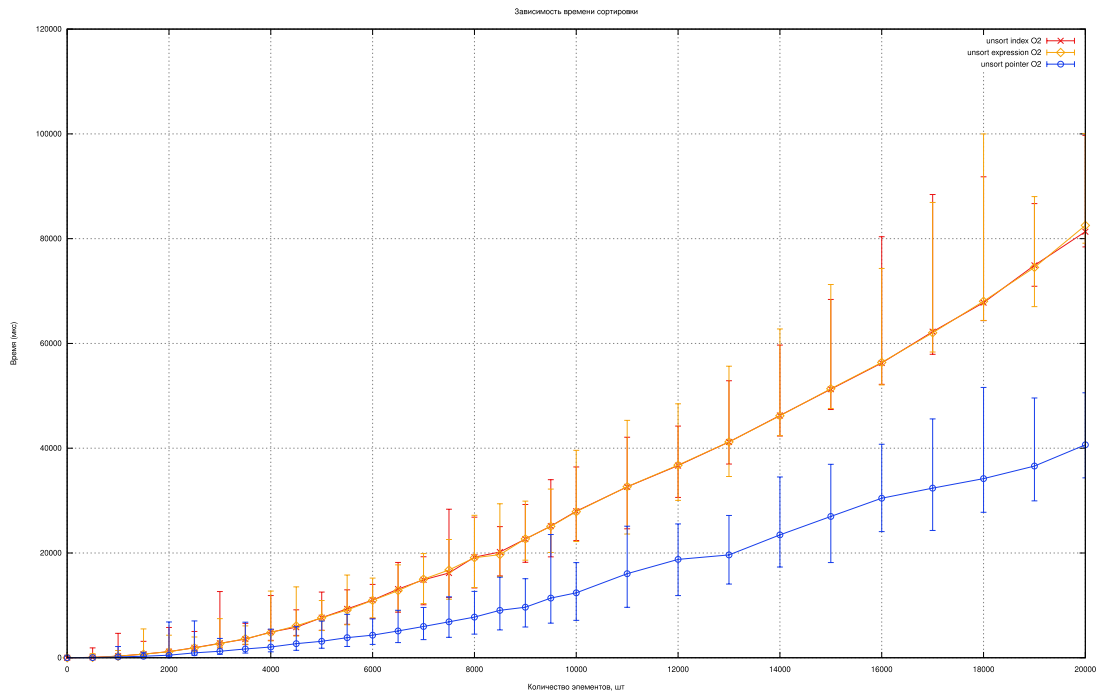
Для сортировки с указателями, оптимизация O2

Длина массива n	Время выполнения $t$ (мкс)	Величина относительной стандартной ошибки среднего	Минимальное время (мкс)	Максимальное время (мкс)
1	0	0,0094	0	1
500	32	0,3536	19	127
1000	159	2,4436	72	2144
1500	284	2,9336	165	1037
2000	488	8,0691	281	6828
2 500	930	9,7781	472	7030
3 000	1235	9,2789	658	3690
3 500	1678	10,4032	895	6789
4 000	2052	14,9197	1126	5462
4 500	2688	17,2024	1420	5928
5 000	3145	20,1694	1811	7001
5 500	3833	24,9800	2155	8287
6 000	4309	29,7007	2550	7417
6500	5112	34,6046	2892	9064
7000	5973	37,8472	3461	9609
7500	6860	41,4386	3893	11632
8000	7760	46,8590	4516	12665
8500	9049	50,0734	5313	15386
9000	9651	57,0888	5869	15088
9500	11390	59,2065	6607	23527
10000	12392	67,5191	7136	18158
11000	16046	62,4809	9629	25116
12000	18779	73,6387	11862	25546
13000	19632	81,5561	14066	27167
14000	23458	67,1960	17316	34510
15000	26977	73,6613	18173	36931
16000	30455	53,8129	24080	40756
17000	32381	84,5815	24297	45592
18000	34196	77,9766	27763	51601
19000	36594	70,3707	29951	49589
20000	40648	62,6859	34336	50579

# График обработки общего случая



# График с ошибкой





## Таблицы для графика с усами:

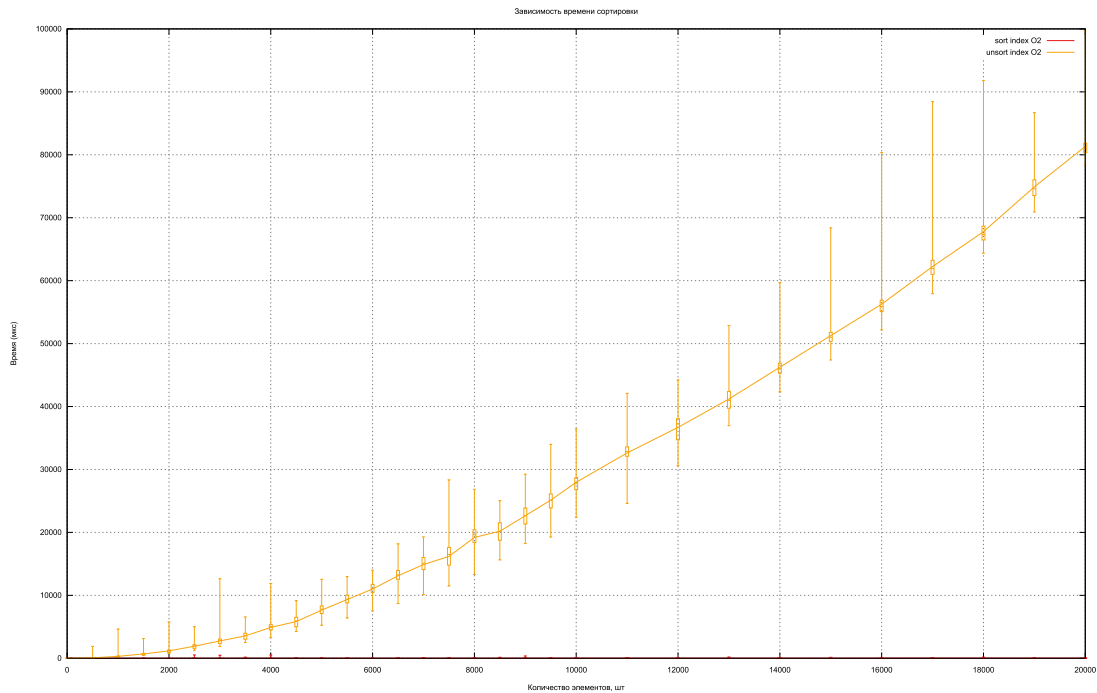
Для сортировки с индексами (наилучший случай), оптимизация O2

Длина массива $n$	Время выполнения $t$ (мкс)	Минимальное время (мкс)	Максимальное время (мкс)	Нижний квартиль (мкс)	Средний квартиль (мкс)	Верхний квартиль (мкс)
1	0	0	1	0	0	0
500	0	0	3	0	1	1
1000	1	0	11	1	1	2
1500	2	1	16	2	2	3
2000	2	1	14	2	3	4
2500	4	2	529	2	4	5
3000	4	2	476	3	5	6
3500	5	3	182	3	6	7
4000	6	3	515	4	7	7
4500	6	3	16	4	7	8
5000	7	4	76	5	8	9
5500	8	4	23	5	9	10
6000	8	5	32	6	8	11
6500	9	5	38	6	11	12
7000	9	5	30	7	11	13
7500	10	5	107	7	11	13
8000	11	6	61	8	13	15
8500	12	7	157	8	15	17
9000	14	7	375	9	16	18
9500	14	7	41	9	16	19
10000	15	7	51	10	17	20
11000	16	8	86	11	18	21
12000	18	9	89	12	21	23
13000	19	9	207	13	22	25
14000	20	10	86	14	22	27
15000	21	11	153	15	23	29
16000	22	12	43	15	24	30
17000	25	12	94	17	28	32
18000	24	13	171	16	26	32
19000	23	14	68	16	23	31
20000	24	14	58	16	29	31

Для сортировки с индексами (общий случай), оптимизация O2

Длина массива $n$	Время выполнения $t$ (мкс)	Минимальное время (мкс)	Максимальное время (мкс)	Нижний квартиль (мкс)	Средний квартиль (мкс)	Верхний квартиль (мкс)
1	0	0	1	0	0	0
500	76	50	1886	59	75	87
1000	306	205	4656	239	303	347
1500	690	464	3136	543	697	777
2000	1170	770	5776	941	1180	1307
2500	1916	1265	5015	1580	1956	2155
3000	2755	1878	12636	2347	2832	3049
3500	3561	2507	6564	3027	3618	3984
4000	4897	3326	11877	4512	5004	5345
4500	5834	4262	9154	5013	5933	6498
5000	7652	5240	12541	7103	7803	8316
5500	9326	6385	12971	8806	9403	10002
6000	11008	7525	14000	10448	11109	11723
6500	13111	8697	18190	12550	13262	13950
7000	14885	10110	19278	14087	15095	16012
7500	16181	11487	28356	14795	16505	17613
8000	19201	13326	26839	18385	19625	20411
8500	20186	15620	25036	18757	20163	21540
9000	22655	18226	29254	21329	22680	23873
9500	25129	19261	33995	23891	25083	26121
10000	27968	22374	36422	26801	27796	28701
11000	32623	24613	42094	32099	32848	33597
12000	36687	30600	44229	34715	37240	38066
13000	41190	36973	52874	39725	40981	42404
14000	46243	42335	59689	45288	46004	46889
15000	51265	47385	68395	50356	50993	51792
16000	56270	52194	80367	55070	55887	56895
17000	62240	57928	88429	60986	61920	63242
18000	67786	64363	91808	66465	67318	68659
19000	74905	70907	86695	73521	74603	76026
20000	81325	78424	99713	80332	81013	81832

## График с усами



### 3. Вопросы по заданию

#### 1) Какой способ обработки быстрее и почему?

Самым быстрым способом обработки является сортировка вставками, реализованная через указатели, с уровнем оптимизации O2. Так как мы исследовали не только различные реализации, но также использовали разные уровни оптимизации, то корректно будет сравнивать скорости работы у исполняемых файлов, которые были собраны с одинаковой оптимизацией. Обработка с помощью реализации через указатели является самой быстрой, так как мы избавляемся от дополнительных вычислений. Которые необходимо производить, например, для обращения к участку памяти которому соответствует индекс или для изменения того же самого индекса.

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

Нет, нельзя, потому что это изменит многие результаты эксперимента и уменьшит точность произведенных измерений. Например, при подсчете среднего медианного будет потеряна точность. И вот простое доказательство:

Возьмем пять измерений: 1 1 1 2 3

Заменяем три одинаковых измерения: 1 2 3

В первом случае среднее медианное равно 1, а во втором мы потеряли точность и среднее медианное стало равно 2.

Схожие последствия ожидают нас при вычислении множества других параметров.

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

Инициализация массива может занимать различное количество времени, которое нельзя узнать точно. Таким образом измерение времени работы всей программы, а не исключительно целевого алгоритма, даст нам результаты с большей погрешностью. В связи с этим мы проводим замеры только у целевого алгоритма.