

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ  
ШКОЛА ЭКОНОМИКИ»

**Контрольное домашнее задание по дисциплине «Построение и анализ  
алгоритмов»**

**Студент:**

Осипов Лев Игоревич, 271ПИ

**Преподаватель:**

Ахметсафина Римма Закиевна, доцент

Москва

2013

# ОГЛАВЛЕНИЕ

Таблицы .....	3
Графики .....	6
Пункт 1 .....	6
Пункт 2 .....	11
Выводы .....	17

# ТАБЛИЦЫ

длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка пузырьком</b>					
1000	7698460	3000970	1493770	1600560	3445640
2000	28101000	11709700	6155750	6861140	13557800
3000	53379400	27160900	13738900	14495700	31781700
4000	120343000	52042000	24758800	24853100	56652800
5000	133844000	85960900	38268800	38173800	89432500
6000	248036000	127474000	54985000	60152800	124576000
7000	235742000	189400000	79467800	75019100	374241000
8000	223933000	341290000	97813500	102585000	225451000
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка пузырьком, Айверсон 1</b>					
1000	7104590	3122230	3175	1660060	3756670
2000	29067800	12332400	5324	5481290	14422200
3000	55474400	28533600	7986	12328900	33023100
4000	120910000	54913800	12123	23724300	61012800
5000	133493000	87671500	16453	34035200	90385100
6000	206763000	130984000	18313	50651100	131536000
7000	215160000	192318000	21616	65984100	249048000
8000	225261000	266823000	24727	108896000	228844000
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка пузырьком, Айверсон 2</b>					
1000	8010770	3316430	4715	2225090	3787040
2000	28261400	12899700	9397	5924970	14928700
3000	69760900	30220000	12219	15962400	34110400
4000	108564000	56271200	16324	23878200	62917300
5000	198725000	97165300	23412	36092300	93370300
6000	205923000	141960000	28255	43464200	273848000
7000	235163000	201250000	29506	73016200	247357000
8000	251928000	271564000	35888	117322000	237770000
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка вставками</b>					
1000	2340970	1037480	5324	18634	2133270
2000	7381660	4094710	10295	47274	8196850
3000	18790000	9178240	15587	96952	18880400
4000	27213400	16212600	20943	94290	33676300
5000	46348600	25740000	25754	124919	50494500
6000	46645200	36370400	31045	150287	110406000
7000	44929800	49537900	37620	87652	128976000
8000	52218200	64604300	41340	129954	129111000

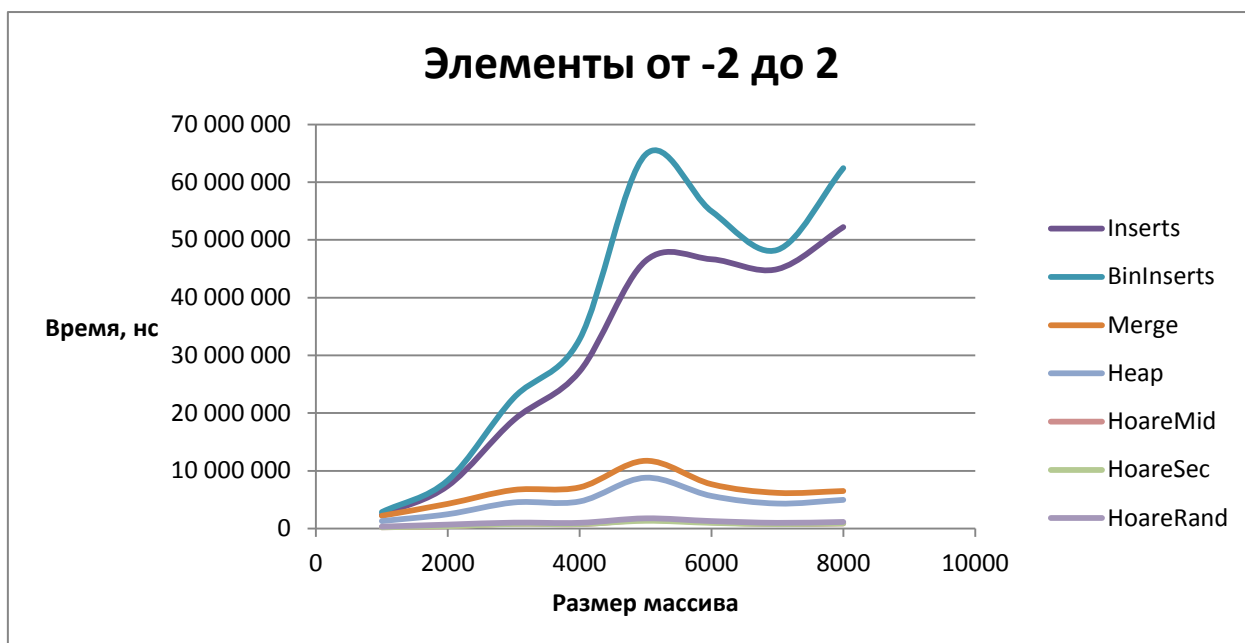
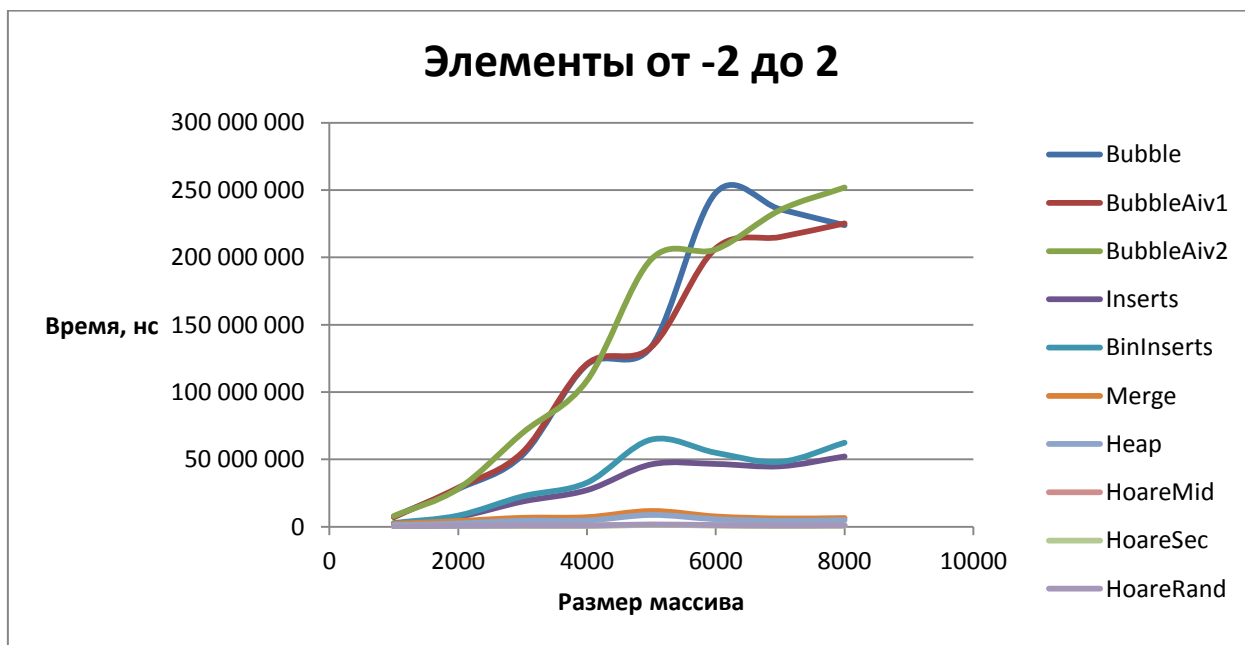
<b>длина</b>	<b>-2 до 2</b>	<b>0 до max</b>	<b>сорт.</b>	<b>почти с.</b>	<b>обратно с.</b>
<b>Сортировка бинарными вставками</b>					
1000	2841930	1046620	5003	89576	1920960
2000	8339540	3762990	9942	182616	7091120
3000	22622500	8568530	14272	380144	15785900
4000	32784200	14342400	19788	457180	28708900
5000	64765500	22022700	24759	355257	44538900
6000	54971600	31902400	29827	760032	83024500
7000	48266500	43316700	34477	932064	109671000
8000	62429300	55701700	39416	937933	111269000
<b>длина</b>	<b>-2 до 2</b>	<b>0 до max</b>	<b>сорт.</b>	<b>почти с.</b>	<b>обратно с.</b>
<b>Сортировка подсчетом</b>					
1000			27165	30083	28127
2000			41084	42912	43425
3000			61802	66581	59365
4000			85663	81590	79345
5000			95541	106446	114656
6000			140730	110647	110454
7000			123476	131975	157311
8000			142237	149101	146984
<b>длина</b>	<b>-2 до 2</b>	<b>0 до max</b>	<b>сорт.</b>	<b>почти с.</b>	<b>обратно с.</b>
<b>Цифровая сортировка (основание 16)</b>					
1000		1052810	780750	776357	768531
2000		2091960	1537740	1570420	1511950
3000		3139420	2323140	2295390	2315950
4000		4210770	3087370	3201420	3226340
5000		5268620	3892620	3784800	4029340
6000		6359860	4889020	4592620	4718850
7000		7388010	5337030	5665190	6840740
8000		8365710	6104180	6149210	6129740
<b>длина</b>	<b>-2 до 2</b>	<b>0 до max</b>	<b>сорт.</b>	<b>почти с.</b>	<b>обратно с.</b>
<b>Сортировка слиянием</b>					
1000	2206780	803040	740597	765773	727864
2000	4272290	1663300	1506150	1517210	1475710
3000	6678910	2530930	2347700	2323170	2213680
4000	7127490	3412260	3091030	3118450	3425210
5000	11730300	4278410	3903330	3883450	3829790
6000	7673960	5230550	4785300	4647650	4631260
7000	6176990	6130350	5611270	5439660	8083060
8000	6494590	7036340	6136320	6186250	6071690

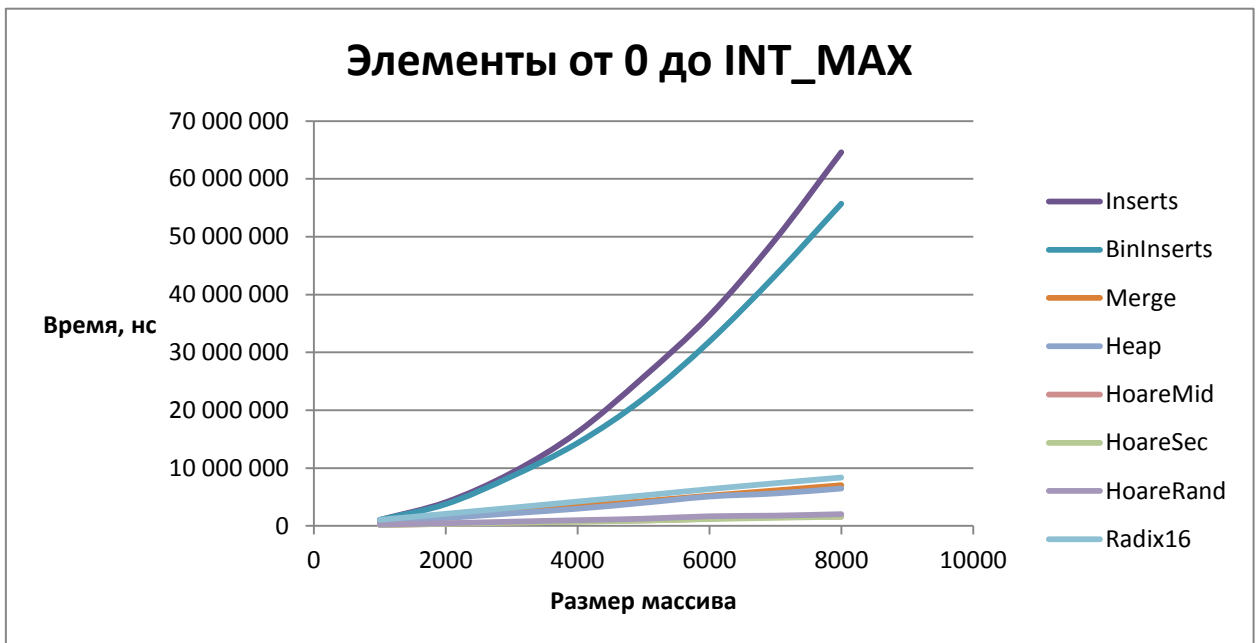
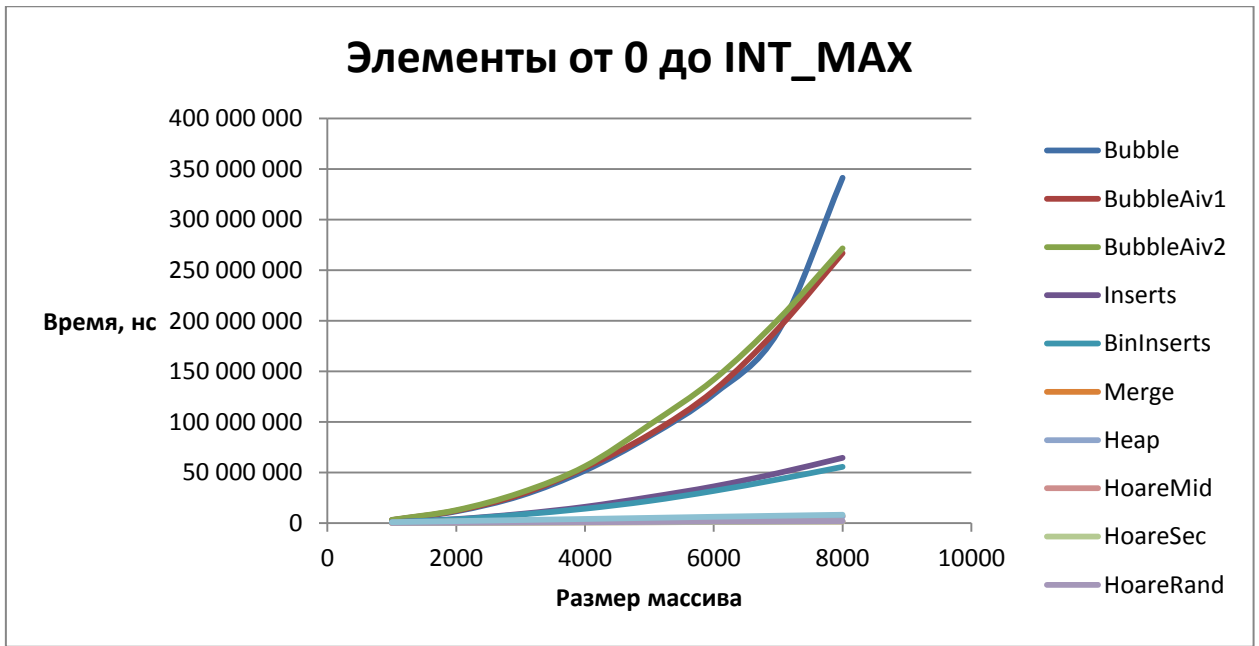
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Пирамидальная сортировка</b>					
1000	1289250	616608	619847	620713	533030
2000	2476370	1361470	1361790	1405340	1170100
3000	4510670	2175700	2154600	2146200	1916500
4000	4685790	2974990	2905080	3042890	3055970
5000	8795080	3990060	3829440	3758370	3370980
6000	5640970	5079050	4736750	4630080	4844700
7000	4325620	5613710	5525900	5459290	6332760
8000	4960860	6463320	6281570	6351680	5790270
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка Хоара по среднему элементу</b>					
1000	216387	155611	69050	64143	62508
2000	393037	361928	146727	166002	151987
3000	821257	554999	269497	253365	268118
4000	750763	762245	360260	361190	374660
5000	1442130	946689	455352	466385	506122
6000	1082700	1245470	548648	543517	556185
7000	753457	1690330	693516	639186	806440
8000	870647	1578210	724593	735433	742297
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка Хоара по второму элементу</b>					
1000	211384	151282	991717	456443	927382
2000	408367	356828	3417130	2297380	2960820
3000	793579	561477	7073930	2660370	5524840
4000	774304	729853	11563400	5838890	8960480
5000	1335620	930877	16820500	10499600	12779400
6000	976034	1186490	23063900	9885810	16130900
7000	763945	1397710	29115000	12012600	25455100
8000	887132	1601620	36265900	19252300	25241300
длина	-2 до 2	0 до max	сорт.	почти с.	обратно с.
<b>Сортировка Хоара по случайному элементу</b>					
1000	325623	224341	148844	150095	157054
2000	677576	461863	329888	324051	316675
3000	1019110	738288	488771	493517	495891
4000	986682	993385	644318	659071	677255
5000	1760220	1246010	821193	794926	814714
6000	1284850	1672180	993385	984405	1032960
7000	979498	1782790	1140340	1106850	1459350
8000	1136740	2035610	1282190	1422370	1324010

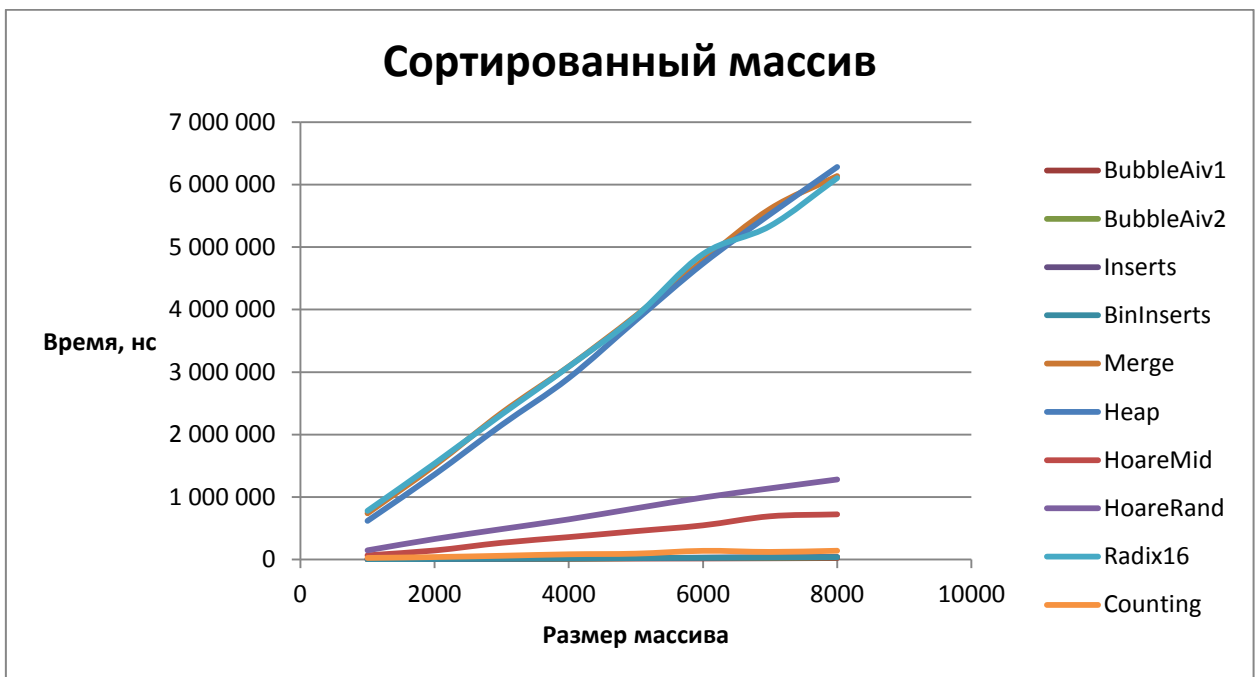
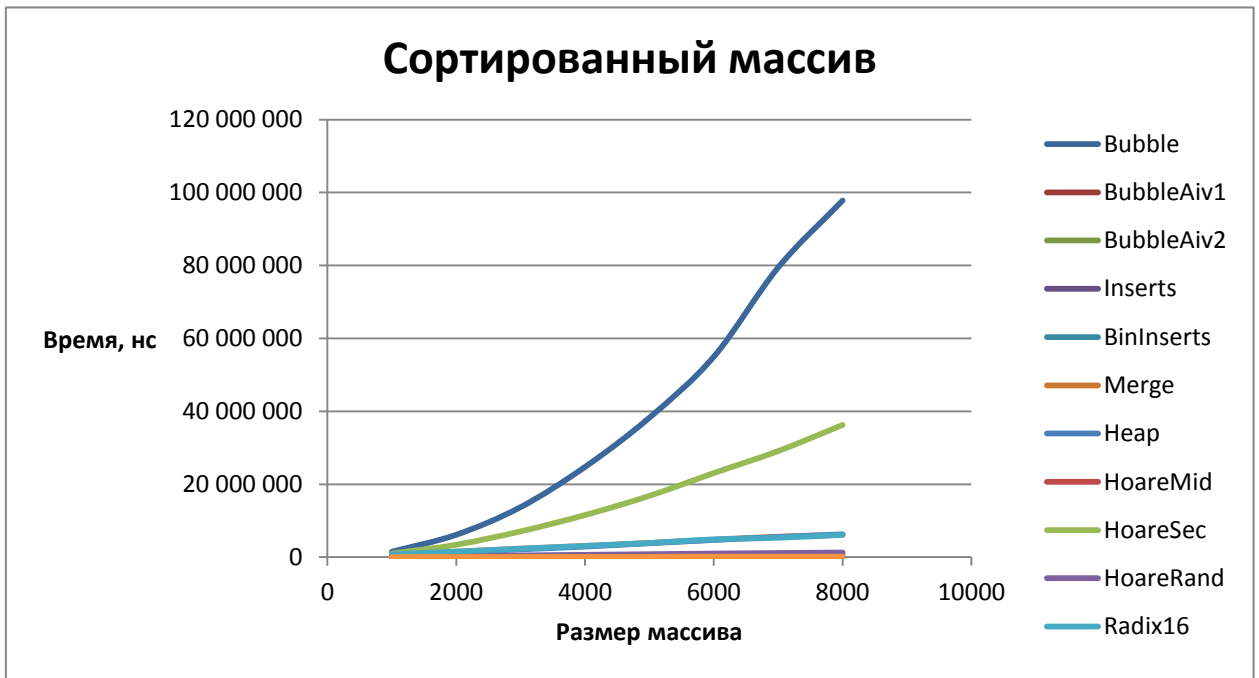
# ГРАФИКИ

## ПУНКТ 1

(каждый график поделен на две части, в одной все сортировки, во второй некоторые исключены для улучшения масштаба)

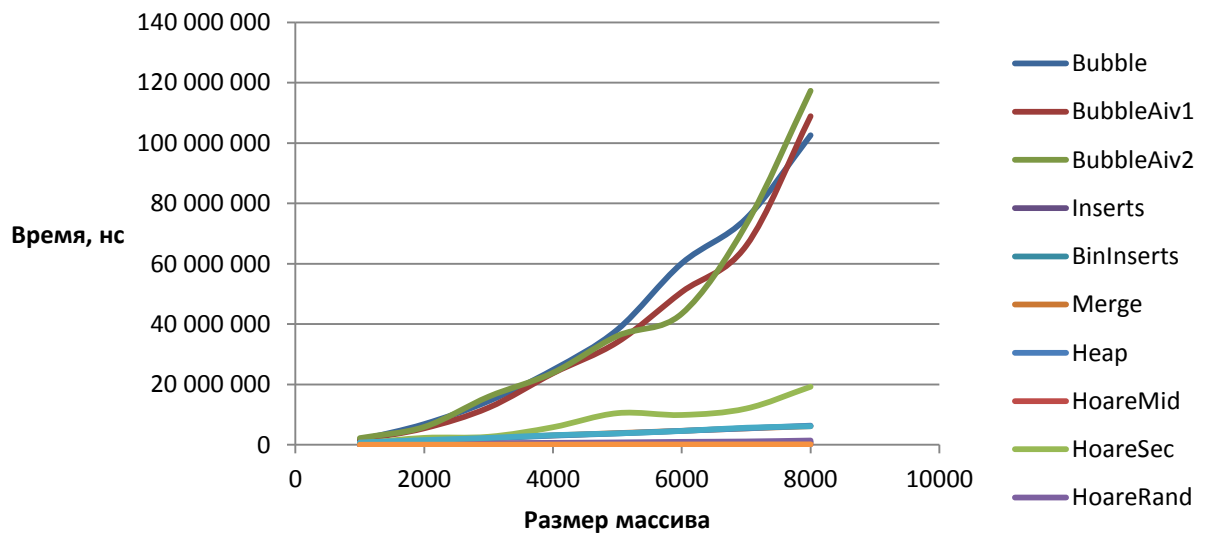




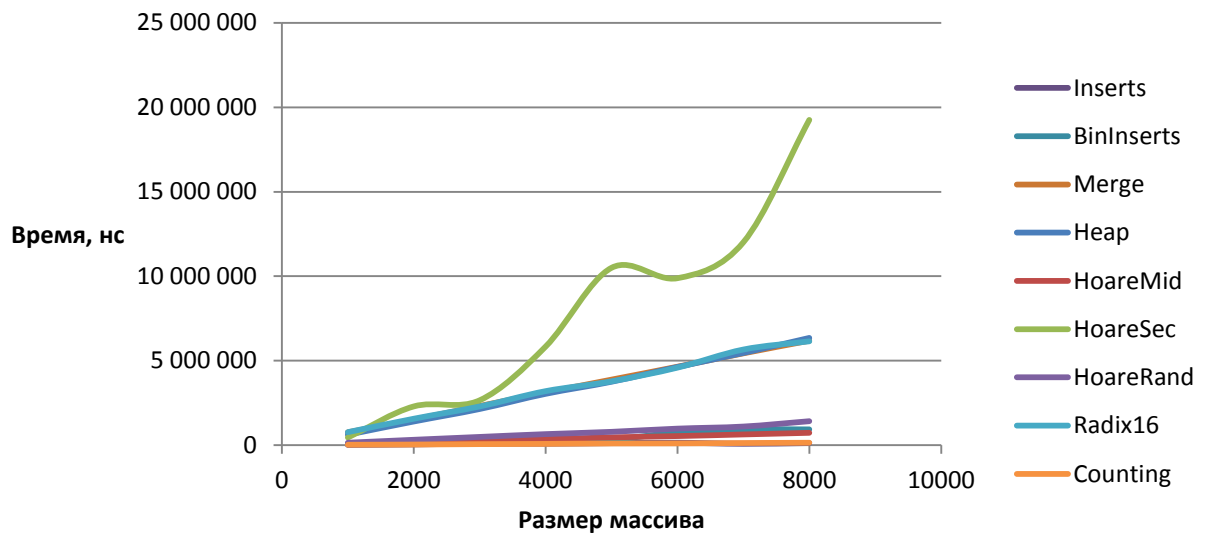




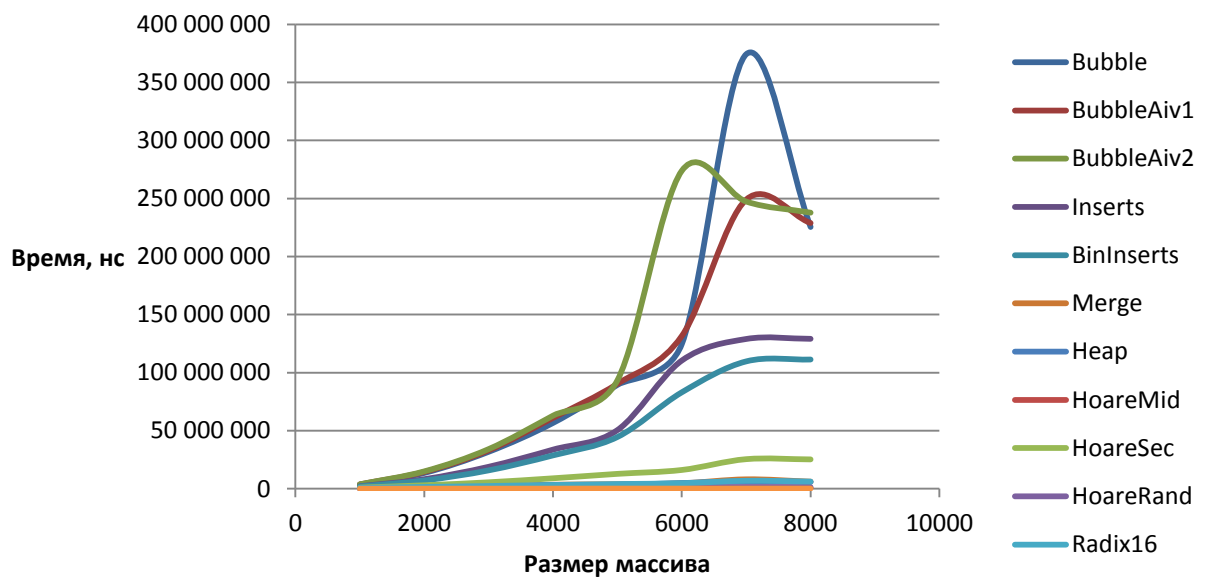
## Почти отсортированный массив



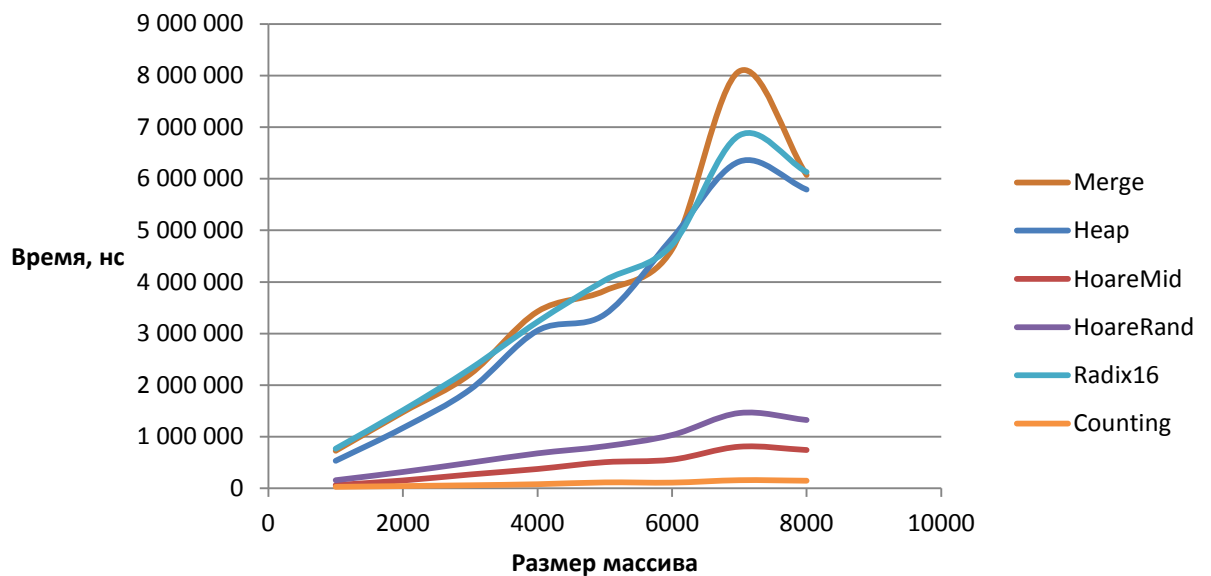
## Почти отсортированный массив



## Обратно сортированный массив

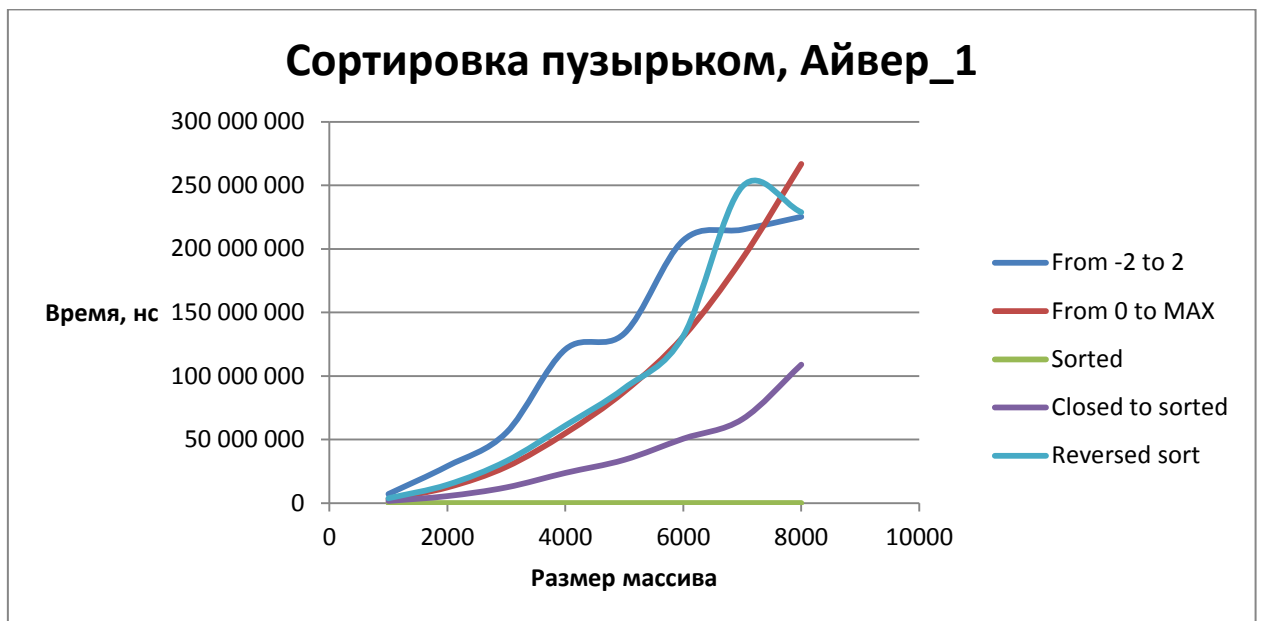
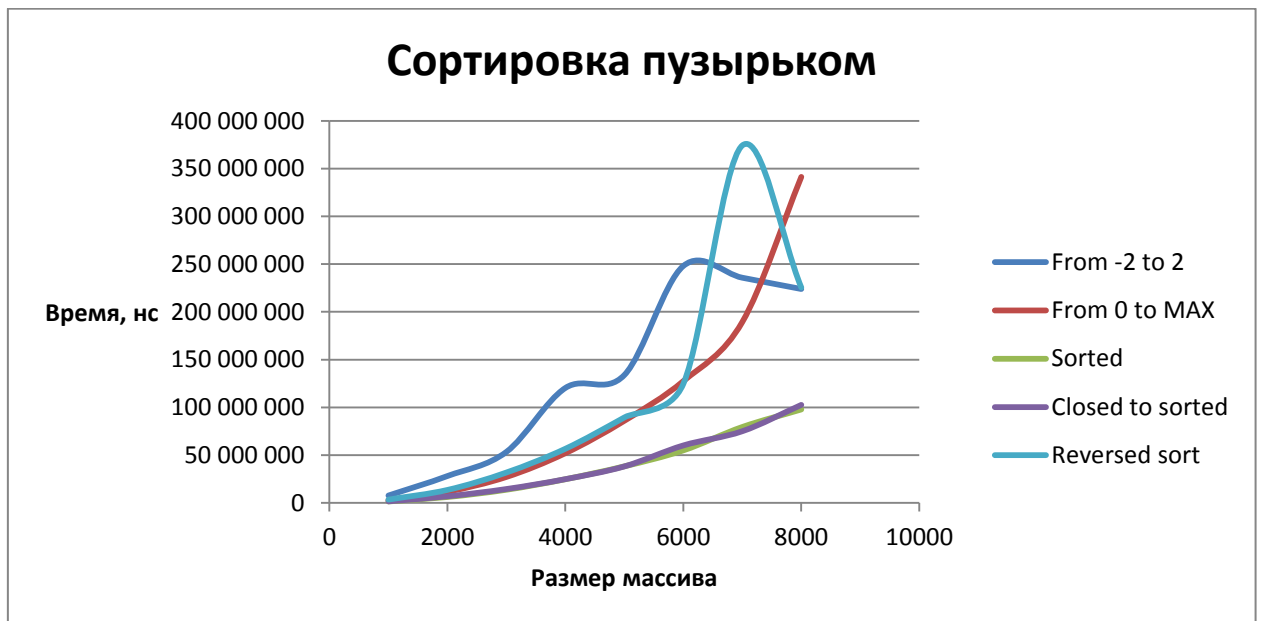


## Обратно сортированный массив

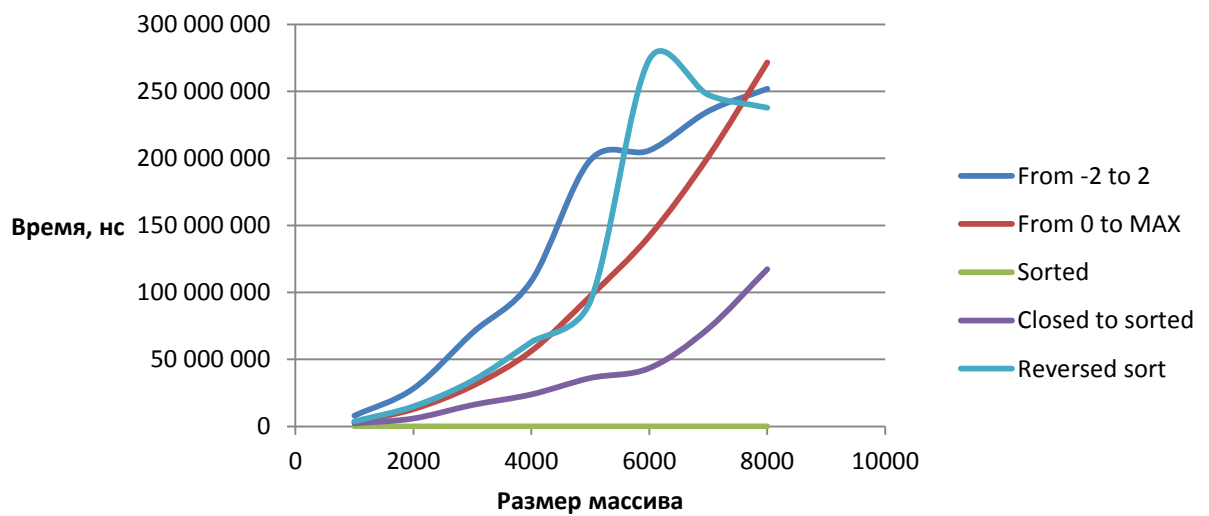


## ПУНКТ 2

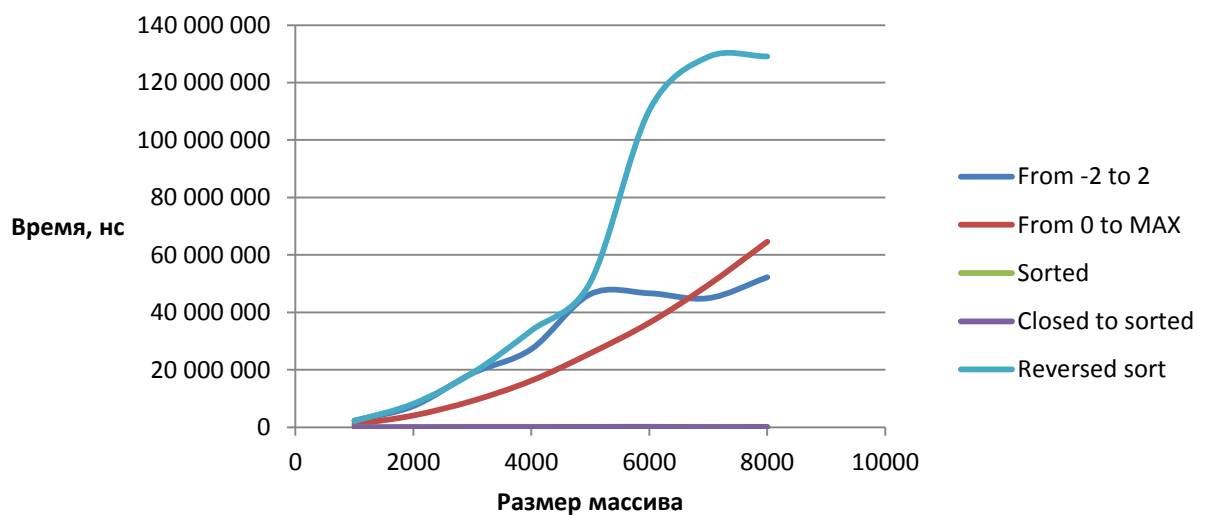
(шкалы для рекурсивных сортировок логарифмические по основанию 2)



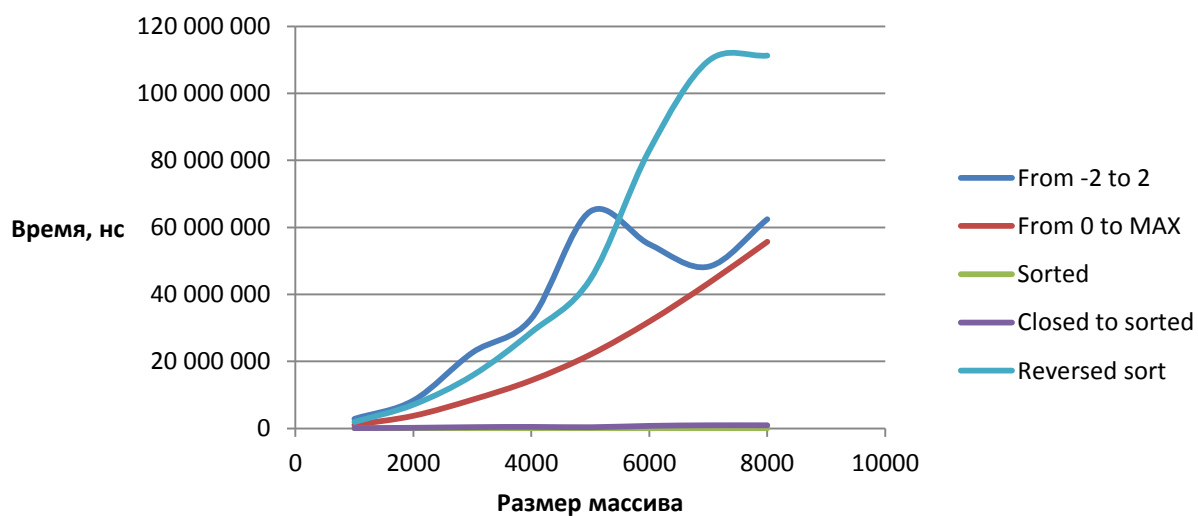
## Сортировка пузырьком, Айверс\_2



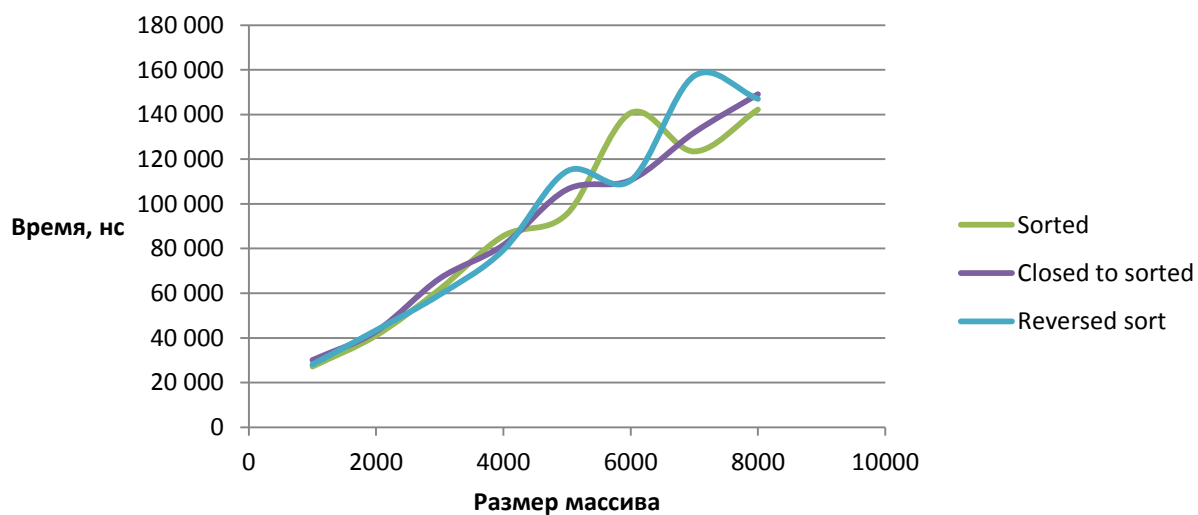
## Сортировка вставками



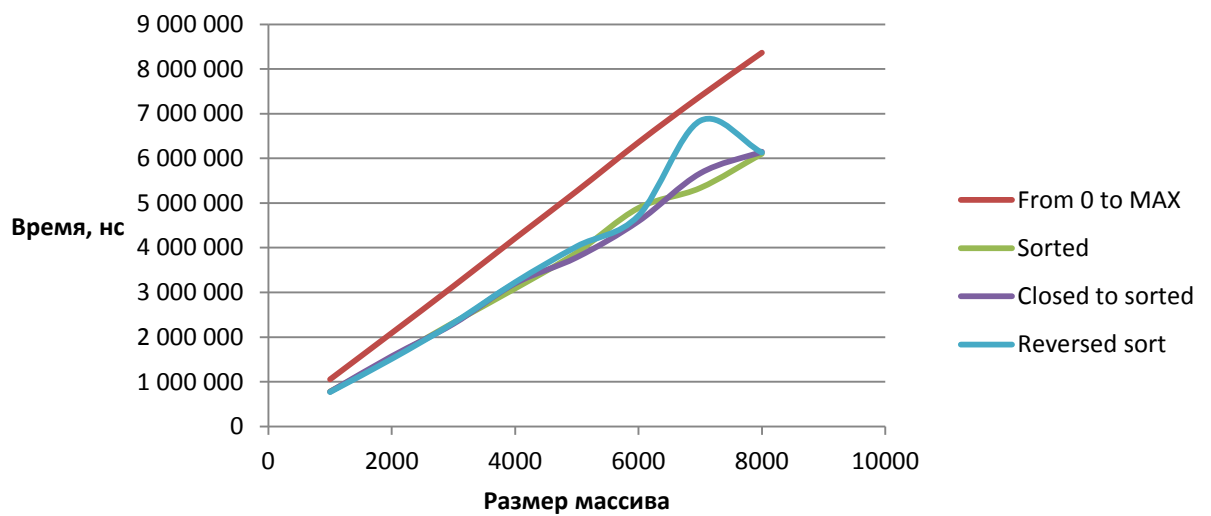
## Сортировка бинарными вставками



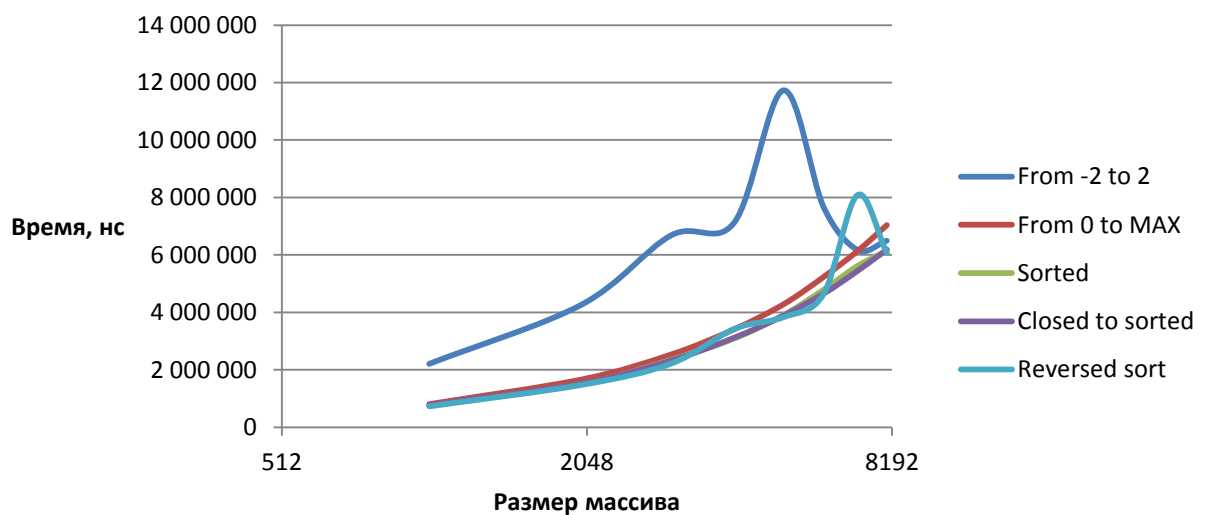
## Сортировка подсчетом



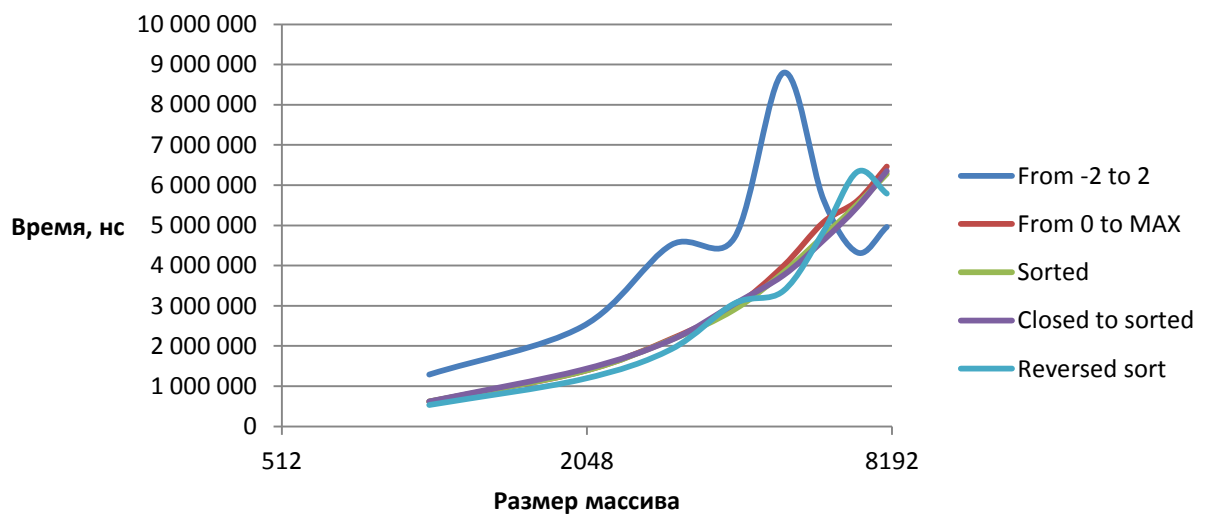
## Цифровая сортировка (основание 16)



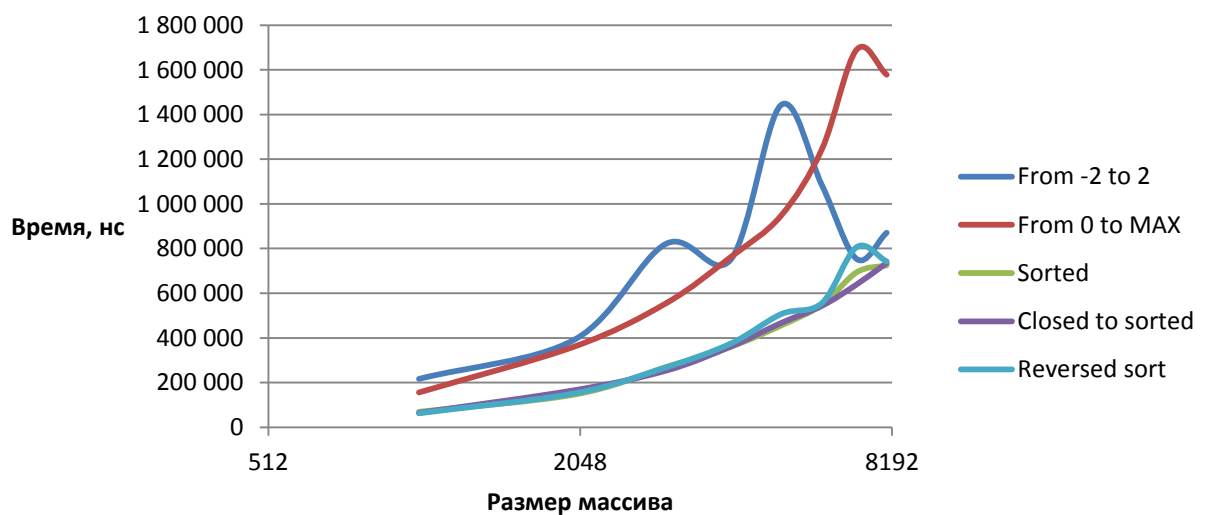
## Сортировка слиянием



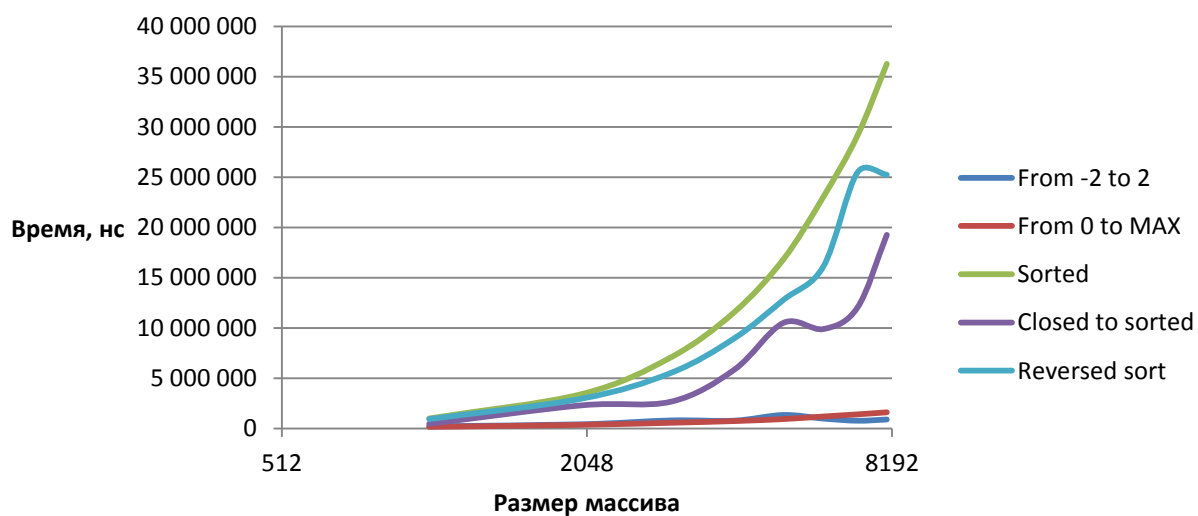
## Пирамидальная сортировка



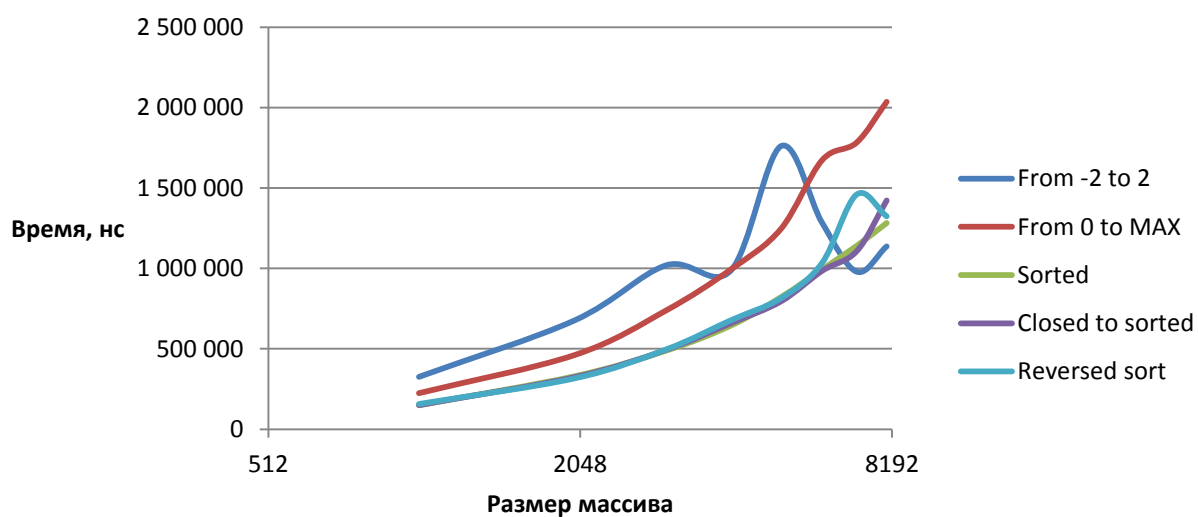
## Сортировка Хоара по среднему элементу



## Сортировка Хоара по второму элементу



## Сортировка Хоара по случайному элементу





## ВЫВОДЫ

В целом быстрее всех работает сортировка подсчетом. Но она не универсальна (например, не может работать на отрицательных числах). Наиболее стабильно сортировка подсчетом работает на массивах, близких к отсортированным.

Сортировка Хоара тоже работает достаточно быстро. Лучшие результаты наблюдаются у сортировки Хоара по среднему элементу, сортировка Хоара по второму элементу работает медленнее остальных.

Достаточно быстро работает цифровая сортировка (основание 16). Интересно отметить, что зависимость времени работы сортировки от размеров массива при элементах массива от 0 до INT\_MAX получилась почти линейной.

Сортировка слиянием работает порядком медленнее на массивах с элементами от -2 до 2. Подобное наблюдается и с пирамидальной сортировкой. В остальных случаях они сравнимы с цифровой сортировкой.

По сравнению с вышеописанными сортировками сортировка вставками и сортировка бинарными вставками работают порядком медленнее, особенно на обратно отсортированных массивах.

Сортировки пузырьком в большинстве случаев работают дольше всех, и между собой они довольно-таки близки, не считая случаев с отсортированным и почти отсортированным массивом, где сортировки пузырьком с условиями Айверсона сильно выигрывают.

При анализе результатов следует учитывать, что среднее из двадцати запусков могло выйти недостаточно точным даже при предварительной прогонке функции.