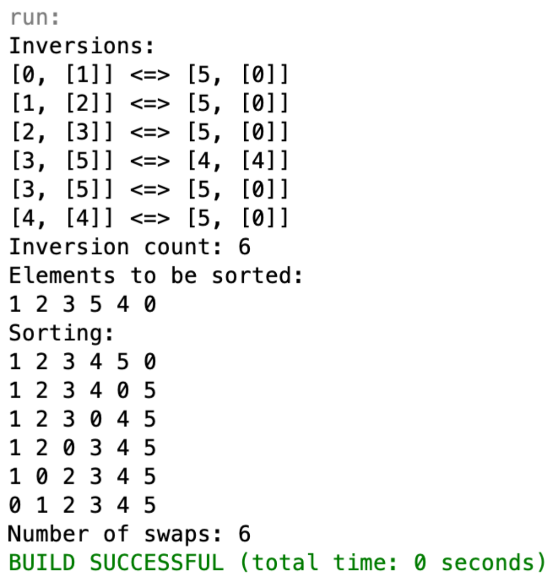
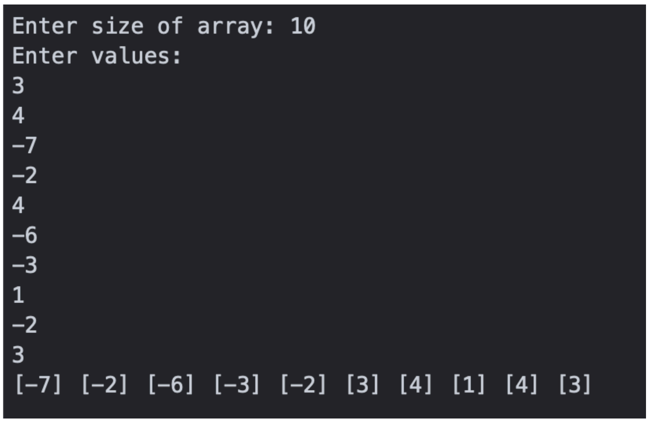
1, 2, 3.



```
run:
Inversions:
[0, [1]] <=> [5, [0]]
[1, [2]] <=> [5, [0]]
[2, [3]] <=> [5, [0]]
[3, [5]] <=> [4, [4]]
[3, [5]] <=> [5, [0]]
[4, [4]] <=> [5, [0]]
Inversion count: 6
Elements to be sorted:
1 2 3 5 4 0
Sorting:
1 2 3 4 5 0
1 2 3 4 0 5
1 2 3 0 4 5
1 2 0 3 4 5
1 0 2 3 4 5
0 1 2 3 4 5
Number of swaps: 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.



```
Enter size of array: 10
Enter values:
3
4
-7
-2
4
-6
-3
1
-2
3
[-7] [-2] [-6] [-3] [-2] [3] [4] [1] [4] [3]
```

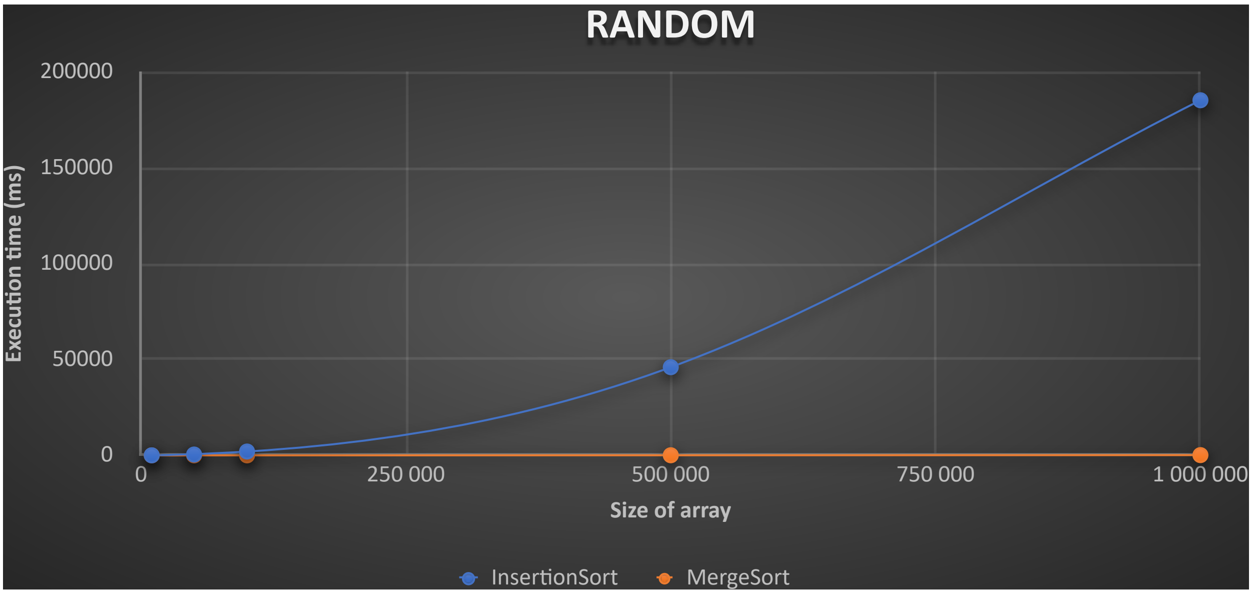
5.

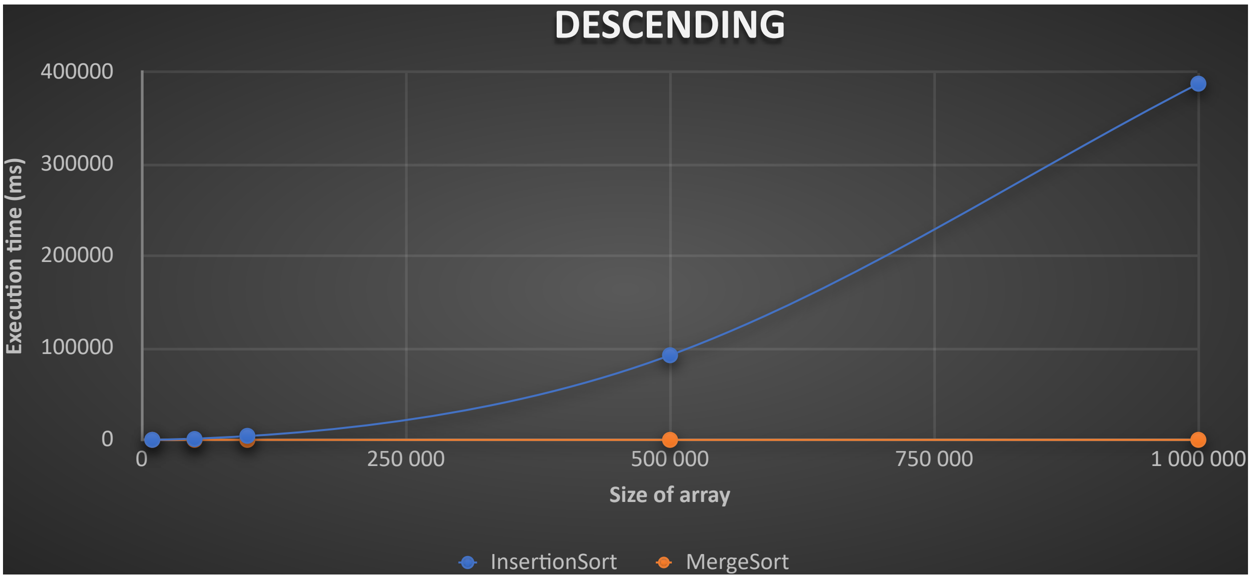
```
OUTPUT SAMPLE:
Inversion sort RANDOM INTS:
Execution time: 34ms
Array size: 10000
Merge sort RANDOM INTS:
Execution time: 5ms
Array size: 10000
Merge sort was faster

Inversion sort RANDOM INTS:
Execution time: 98ms
Array size: 20000
Merge sort RANDOM INTS:
Execution time: 5ms
Array size: 20000
Merge sort was faster

Inversion sort RANDOM INTS:
Execution time: 406ms
Array size: 40000
Merge sort RANDOM INTS:
Execution time: 6ms
Array size: 40000
Merge sort was faster
```

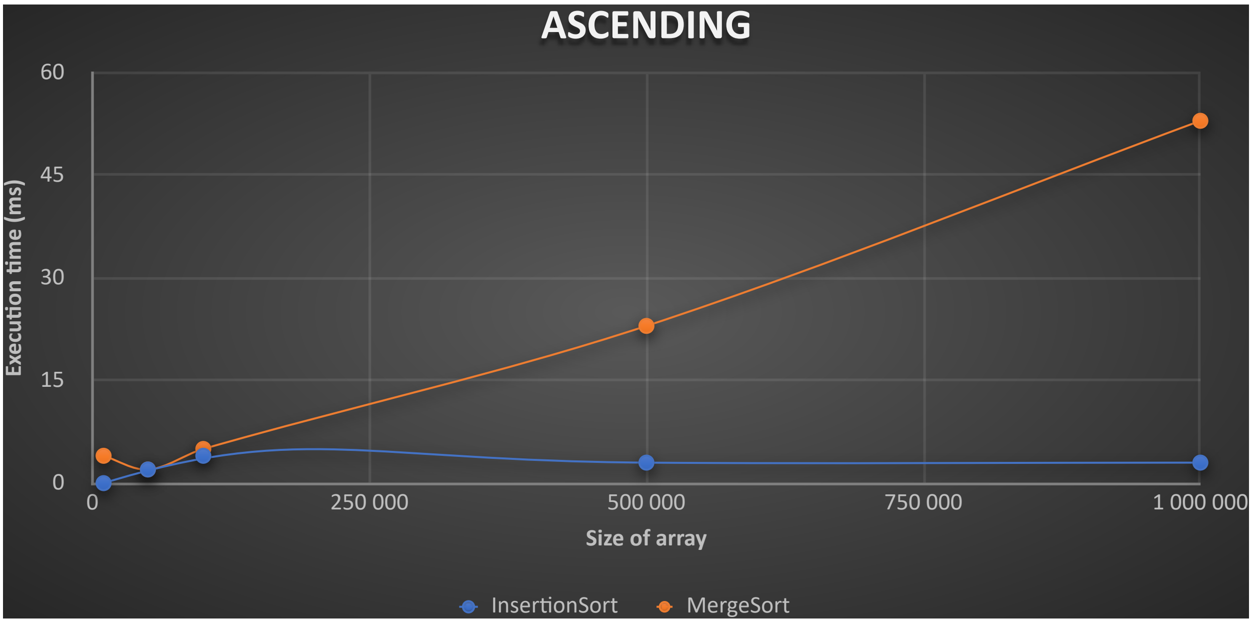
```
Inversion sort RANDOM INTS:
Execution time: 1357ms
Array size: 80000
Merge sort RANDOM INTS:
Execution time: 10ms
Array size: 80000
Merge sort was faster

Inversion sort RANDOM INTS:
Execution time: 6282ms
Array size: 160000
Merge sort RANDOM INTS:
Execution time: 20ms
Array size: 160000
Merge sort was faster

BUILD SUCCESSFUL (total time: 8 seconds)
```

Examples:

	RANDOM		**DESCENDING**		**ASCENDING**	
Size	InsertionSort	MergeSort	InsertionSort	MergeSort	InsertionSort	MergeSort
10 000	28	2	51	6	0	4
50 000	452	6	938	2	2	2
100 000	2 016	12	4 394	5	4	5
500 000	46 034	65	92 161	21	3	23
1 000 000	185 375	139	387 846	45	3	53


RANDOM
Execution time (ms)
200000
150000
100000
50000
0
0
250 000
500 000
750 000
1 000 000
Size of array
InsertionSort
MergeSort


DESCENDING
Execution time (ms)
400000
300000
200000
100000
0
0
250 000
500 000
750 000
1 000 000
Size of array
InsertionSort
MergeSort

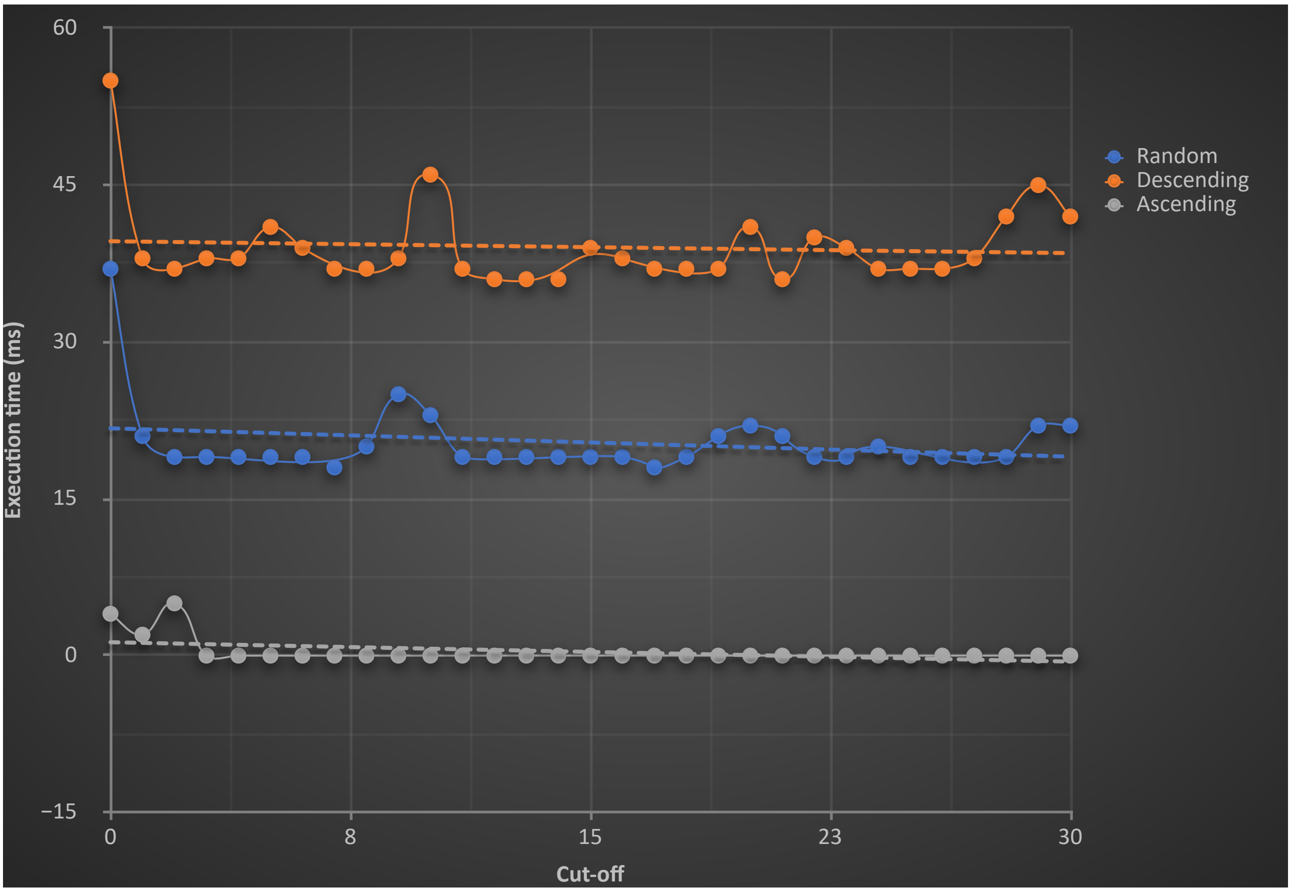

ASCENDING
Execution time (ms)
60
45
30
15
0
0
250 000
500 000
750 000
1 000 000
Size of array
InsertionSort
MergeSort

6.

Size: 20.000 ASCEDNING ORDER Cut-off range 0-30

```
OUTPUT SAMPLE:
Size of array:
20000
CUTOFF: 0
Execution time: 105ms
Array size: 20000

CUTOFF: 1
Execution time: 75ms
Array size: 20000

CUTOFF: 2
Execution time: 82ms
Array size: 20000

CUTOFF: 3
Execution time: 77ms
Array size: 20000

CUTOFF: 4
Execution time: 78ms
Array size: 20000

CUTOFF: 5
Execution time: 78ms
Array size: 20000

CUTOFF: 6
Execution time: 77ms
Array size: 20000

CUTOFF: 7
Execution time: 74ms
Array size: 20000

CUTOFF: 8
Execution time: 80ms
Array size: 20000

CUTOFF: 9
Execution time: 89ms
Array size: 20000

CUTOFF: 10
Execution time: 95ms
Array size: 20000

CUTOFF: 11
Execution time: 135ms
Array size: 20000

CUTOFF: 12
Execution time: 87ms
Array size: 20000

CUTOFF: 13
Execution time: 76ms
Array size: 20000

CUTOFF: 14
Execution time: 84ms
Array size: 20000

CUTOFF: 15
Execution time: 102ms
Array size: 20000

CUTOFF: 16
Execution time: 82ms
Array size: 20000

CUTOFF: 17
Execution time: 108ms
Array size: 20000

CUTOFF: 18
Execution time: 78ms
Array size: 20000

CUTOFF: 19
Execution time: 89ms
Array size: 20000

CUTOFF: 20
Execution time: 105ms
Array size: 20000

CUTOFF: 21
Execution time: 95ms
Array size: 20000

CUTOFF: 22
Execution time: 90ms
Array size: 20000

CUTOFF: 23
Execution time: 131ms
Array size: 20000

CUTOFF: 24
Execution time: 101ms
Array size: 20000

CUTOFF: 25
Execution time: 81ms
Array size: 20000

CUTOFF: 26
Execution time: 84ms
Array size: 20000

CUTOFF: 27
Execution time: 85ms
Array size: 20000

CUTOFF: 28
Execution time: 83ms
Array size: 20000

CUTOFF: 29
Execution time: 82ms
Array size: 20000

CUTOFF: 30
Execution time: 84ms
Array size: 20000

BUILD SUCCESSFUL (total
time: 7 seconds)
```

	Array size: 10 000		
Cut-off	Random	Descending	Ascending
0	37	55	4
1	21	38	2
2	19	37	5
3	19	38	0
4	19	38	0
5	19	41	0
6	19	39	0
7	18	37	0
8	20	37	0
9	25	38	0
10	23	46	0
11	19	37	0
12	19	36	0
13	19	36	0
14	19	36	0
15	19	39	0
16	19	38	0
17	18	37	0
18	19	37	0
19	21	37	0
20	22	41	0
21	21	36	0
22	19	40	0
23	19	39	0
24	20	37	0
25	19	37	0
26	19	37	0
27	19	38	0
28	19	42	0
29	22	45	0
30	22	42	0


60
45
30
15
0
−15
Execution time (ms)
0
8
15
23
30
Cut-off
Random
Descending
Ascending