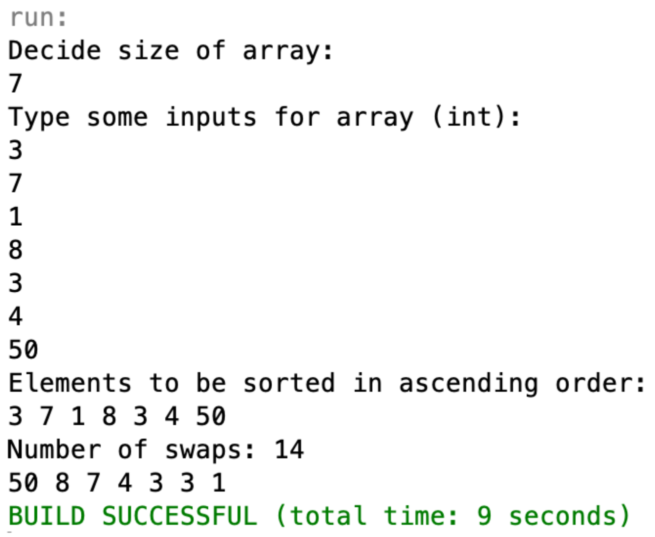
HG1.



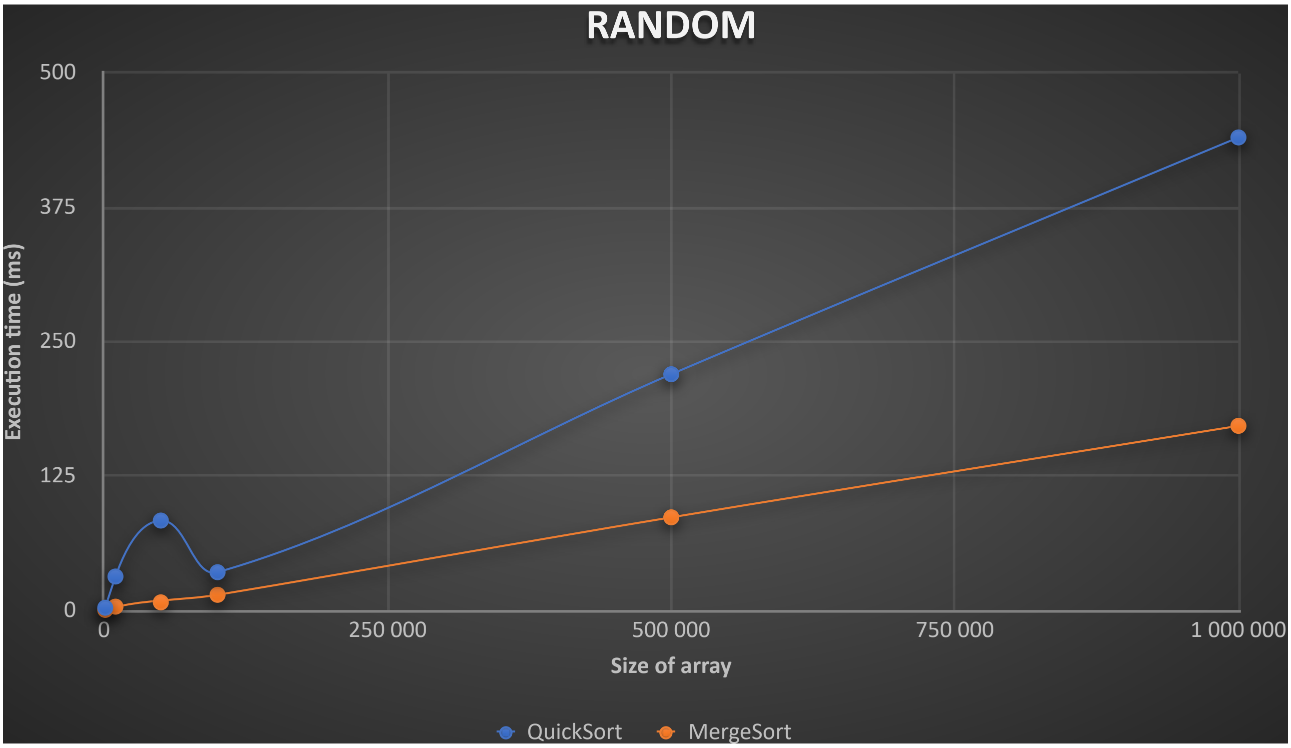
```
run:
Decide size of array:
7
Type some inputs for array (int):
3
7
1
8
3
4
50
Elements to be sorted in ascending order:
3 7 1 8 3 4 50
Number of swaps: 14
50 8 7 4 3 3 1
BUILD SUCCESSFUL (total time: 9 seconds)
```

HG2.

```
OUTPUT SAMPLE
run:
Size:
10000
Merge sort RANDOM INTS:
Execution time: 5ms
Array size: 10000
Quick sort RANDOM INTS:
Execution time: 19ms
Array size: 10000
Merge sort was faster

Size:
50000
Merge sort RANDOM INTS:
Execution time: 6ms
Array size: 50000
Quick sort RANDOM INTS:
Execution time: 21ms
Array size: 50000
Merge sort was faster

Size:
100000
Merge sort RANDOM INTS:
Execution time: 16ms
Array size: 100000
Quick sort RANDOM INTS:
Execution time: 40ms
Array size: 100000
Merge sort was faster
```

```
Size:
500000
Merge sort RANDOM INTS:
Execution time: 78ms
Array size: 500000
Quick sort RANDOM INTS:
Execution time: 208ms
Array size: 500000
Merge sort was faster

Size:
1000000
Merge sort RANDOM INTS:
Execution time: 165ms
Array size: 1000000
Quick sort RANDOM INTS:
Execution time: 668ms
Array size: 1000000
Merge sort was faster

BUILD SUCCESSFUL (total time: 29 seconds)
```

Size	QuickSort	MergeSort
1 000	3	1
10 000	32	4
50 000	84	8
100 000	36	15
500 000	220	87
1 000 000	440	172

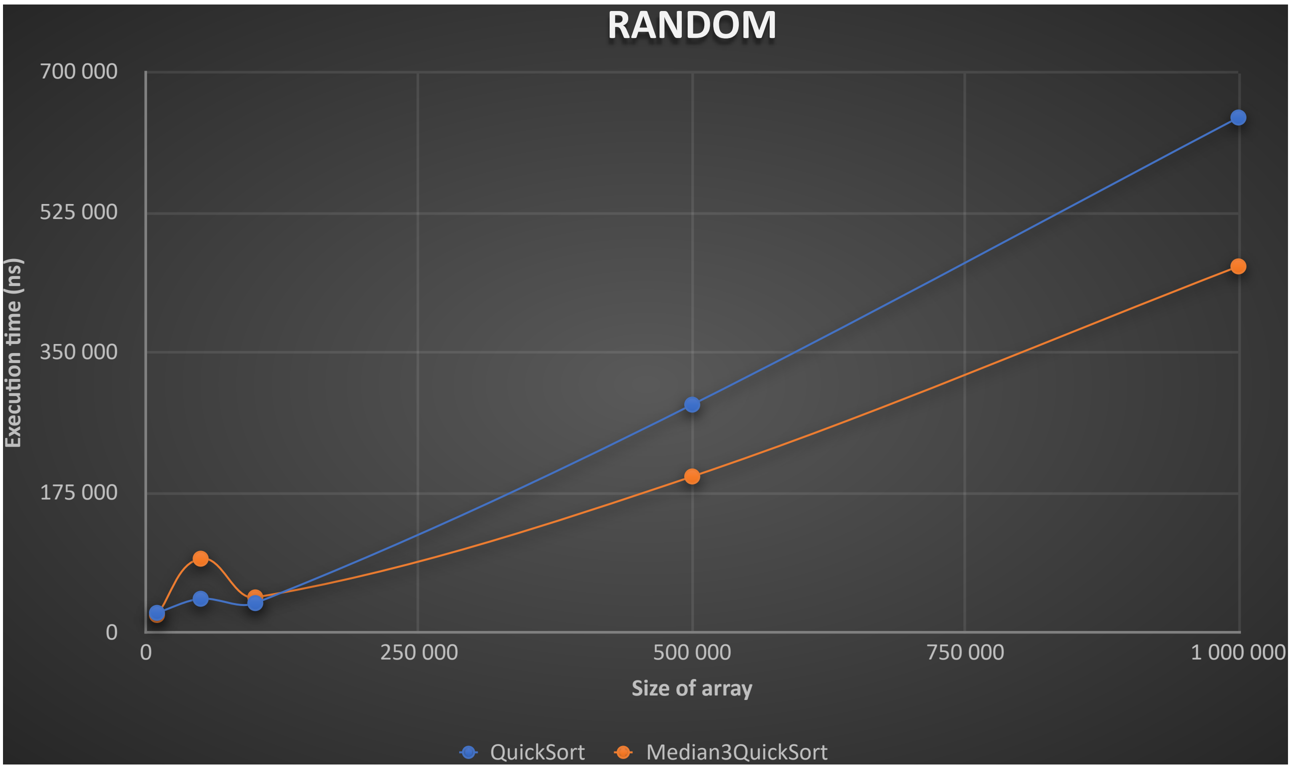


HG3.

OUTPUT SAMPLE

```
run:

Size:
10000
Quick sort first element partitioning RANDOM
INTS:
Execution time: 25ms
Array size: 10000
Quick sort median-of-three partitioning
RANDOM INTS:
Execution time: 23ms
Array size: 10000
Quick sort median-of-three partitioning was
faster

Size:
50000
Quick sort first element partitioning RANDOM
INTS:
Execution time: 43ms
Array size: 50000
Quick sort median-of-three partitioning
RANDOM INTS:
Execution time: 93ms
Array size: 50000
Quick sort first element partitioning was faster

Size:
100000
Quick sort first element partitioning RANDOM
INTS:
Execution time: 38ms
Array size: 100000
Quick sort median-of-three partitioning
RANDOM INTS:
Execution time: 45ms
Array size: 100000
Quick sort first element partitioning was faster

Size:
500000
Quick sort first element partitioning RANDOM
INTS:
Execution time: 285ms
Array size: 500000
Quick sort median-of-three partitioning
RANDOM INTS:
Execution time: 196ms
Array size: 500000
Quick sort median-of-three partitioning was
faster

Size:
1000000
Quick sort first element partitioning RANDOM
INTS:
Execution time: 644ms
Array size: 1000000
Quick sort median-of-three partitioning
RANDOM INTS:
Execution time: 458ms
Array size: 1000000
Quick sort median-of-three partitioning was
faster

BUILD SUCCESSFUL (total time: 26 seconds)
```

Size	QuickSort	Median3QuickSort
10 000	25 873	23 230
50 000	43 471	93 472
100 000	38 046	45 195
500 000	285 739	196 085
1 000 000	644 046	458 258


RANDOM
Execution time (ns)
Size of array
QuickSort
Median3QuickSort