Contents

1 Olav's tests **3**

1.1 Bubblesort . . . 3

1.1.1 Random . . . 3

1.1.2 Sorted . . . 4

1.1.3 Reverse Sorted . . . 5

1.1.4 Uniquely Random . . . 6

1.1.5 Almost Sorted . . . 7

1.1.6 Comparison . . . 7

1.2 Heapsort . . . 8

1.2.1 Random . . . 8

1.2.2 Sorted . . . 9

1.2.3 Reverse Sorted . . . 10

1.2.4 Uniquely Random . . . 11

1.2.5 Almost Sorted . . . 12

1.2.6 Comparison . . . 12

1.3 Quicksort . . . 13

1.3.1 Random . . . 13

1.3.2 Sorted . . . 14

1.3.3 Reverse Sorted . . . 15

1.3.4 Uniquely Random . . . 16

1.3.5 Almost Sorted . . . 17

1.3.6 Comparison - Random, Uniquely Random and Almost Sorted . . . 17

1.3.7 Comparison - Sorted and Reverse Sorted . . . 18

1.4 std::qsort . . . 18

1.4.1 Random . . . 18

1.4.2 Sorted . . . 19

1.4.3 Reverse Sorted . . . 20

1.4.4 Uniquely Random . . . 21

1.4.5 Almost Sorted . . . 22

1.4.6 Comparison . . . 22

1.5 Comparison . . . 23

1.5.1 Random . . . 23

1.5.2 Sorted - Heapsort and std::qsort . . . 23

1.5.3 Sorted - Bubblesort and Quicksort . . . 24

1.5.4 Reverse Sorted - Heapsort and std::qsort . . . 25

1.5.5 Reverse Sorted - Bubblesort and Quicksort . . . 26

1.5.6 Uniquely Random . . . 26

1.5.7 Almost Sorted . . . 27

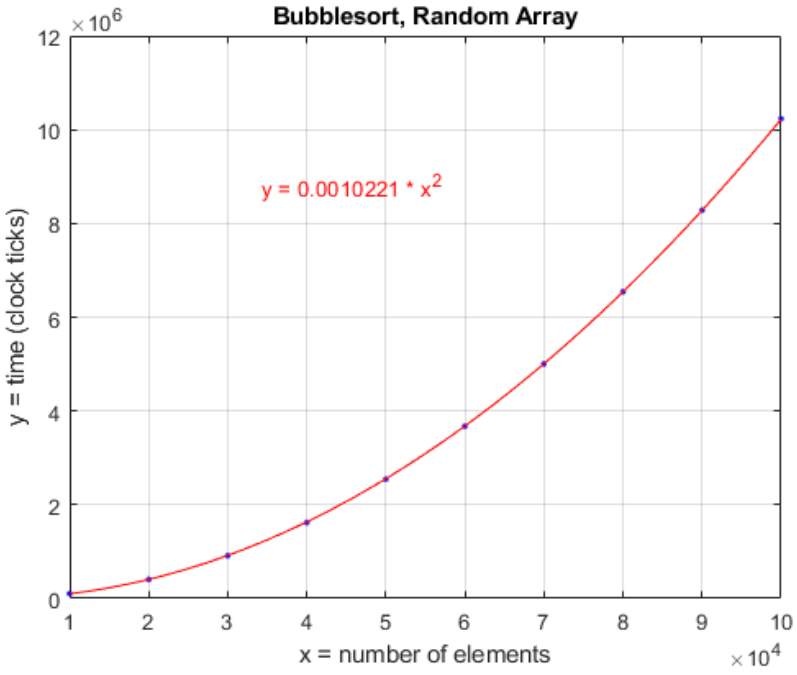
2 Oscar's tests 1 **28**
2.1 Bubblesort . . . 28
2.1.1 Random . . . 28
2.1.2 Uniquely Random . . . 29
2.1.3 Almost Sorted . . . 30
2.1.4 Comparison . . . 30
2.2 Heapsort . . . 31
2.2.1 Random . . . 31
2.2.2 Uniquely Random . . . 32
2.2.3 Almost Sorted . . . 33
2.2.4 Comparison . . . 33
2.3 Quicksort . . . 34
2.3.1 Random . . . 34
2.3.2 Uniquely Random . . . 35
2.3.3 Almost Sorted . . . 36
2.3.4 Comparison . . . 36
2.4 std::qsort . . . 37
2.4.1 Random . . . 37
2.4.2 Uniquely Random . . . 38
2.4.3 Almost Sorted . . . 39
2.4.4 Comparison . . . 39
2.5 Comparison . . . 40
2.5.1 Random . . . 40
2.5.2 Uniquely Random . . . 40
2.5.3 Almost Sorted . . . 41

3 Oscar's tests 2 **41**
3.1 Heapsort . . . 41
3.1.1 Random . . . 41
3.1.2 Uniquely Random . . . 42
3.1.3 Almost Sorted . . . 43
3.1.4 Comparison . . . 43
3.2 Quicksort . . . 44
3.2.1 Random . . . 44
3.2.2 Uniquely Random . . . 45
3.2.3 Almost Sorted . . . 46
3.2.4 Comparison . . . 46
3.3 std::qsort . . . 47
3.3.1 Random . . . 47
3.3.2 Uniquely Random . . . 48
3.3.3 Almost Sorted . . . 49
3.3.4 Comparison . . . 49
3.4 Comparison . . . 50
3.4.1 Random . . . 50
3.4.2 Uniquely Random . . . 50
3.4.3 Almost Sorted . . . 51

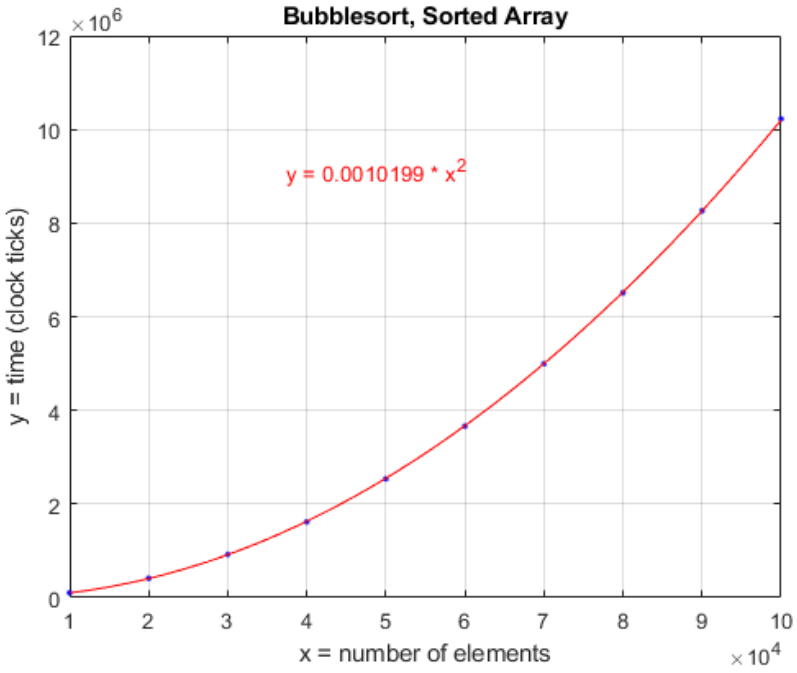
1 Olav's tests

1.1 Bubblesort

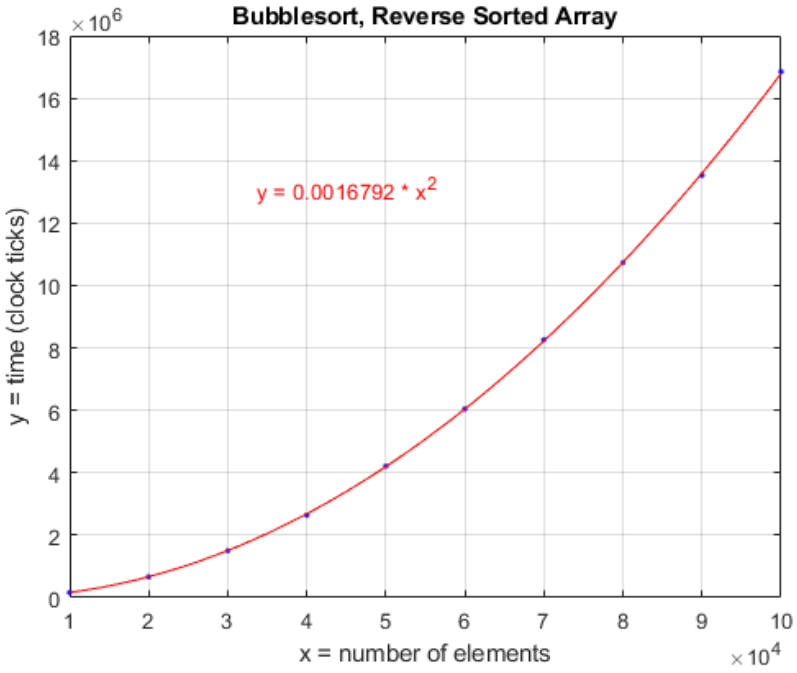
1.1.1 Random



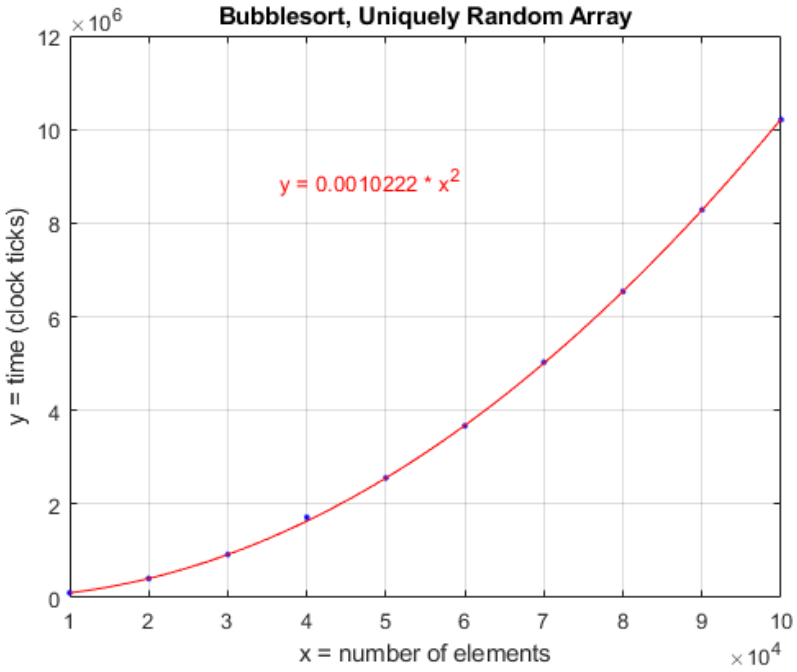
1.1.2 Sorted


Bubblesort, Sorted Array
×10^6
12
10
8
6
4
2
0
y = time (clock ticks)
y = 0.0010199 * x^2
1
2
3
4
5
6
7
8
9
10
x = number of elements
×10^4

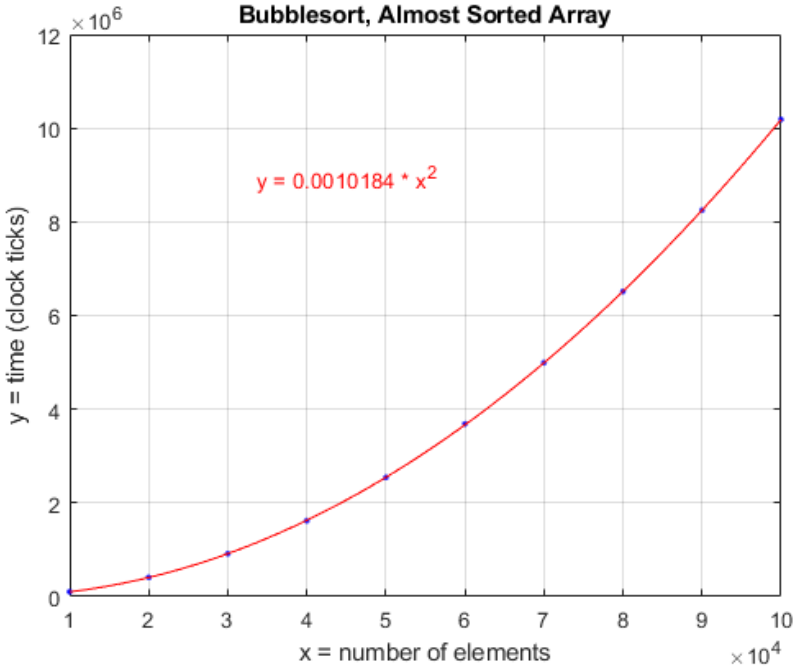
1.1.3 Reverse Sorted



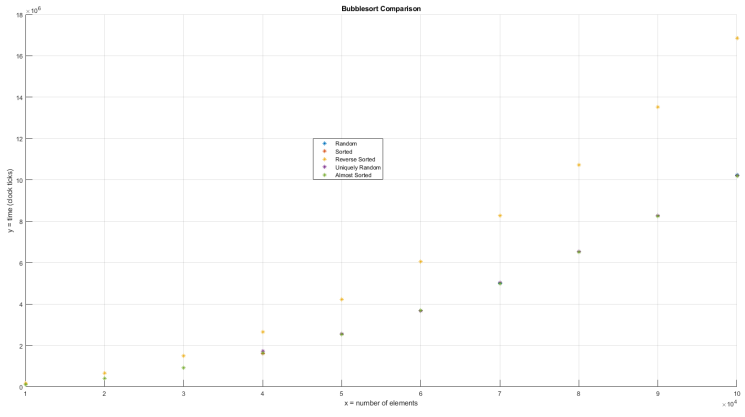
1.1.4 Uniquely Random



1.1.5 Almost Sorted

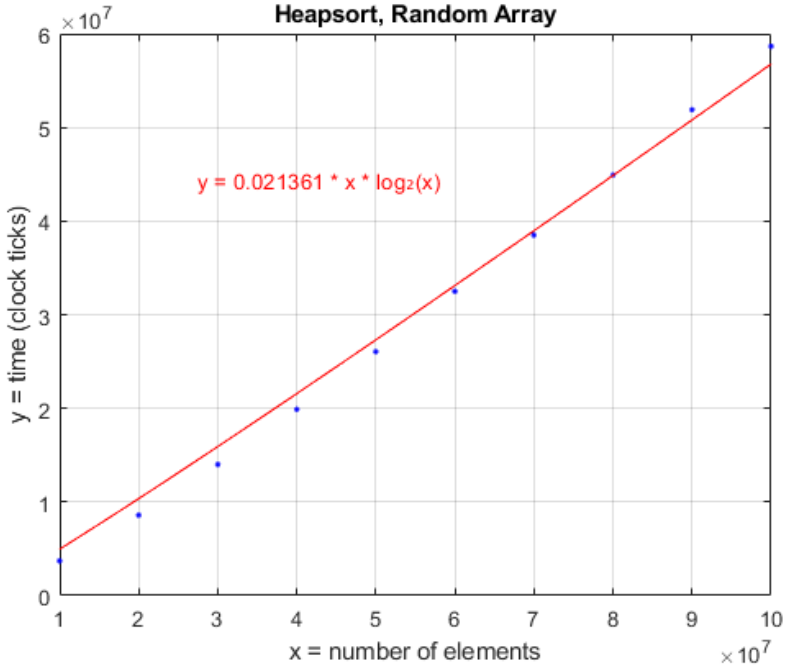


1.1.6 Comparison

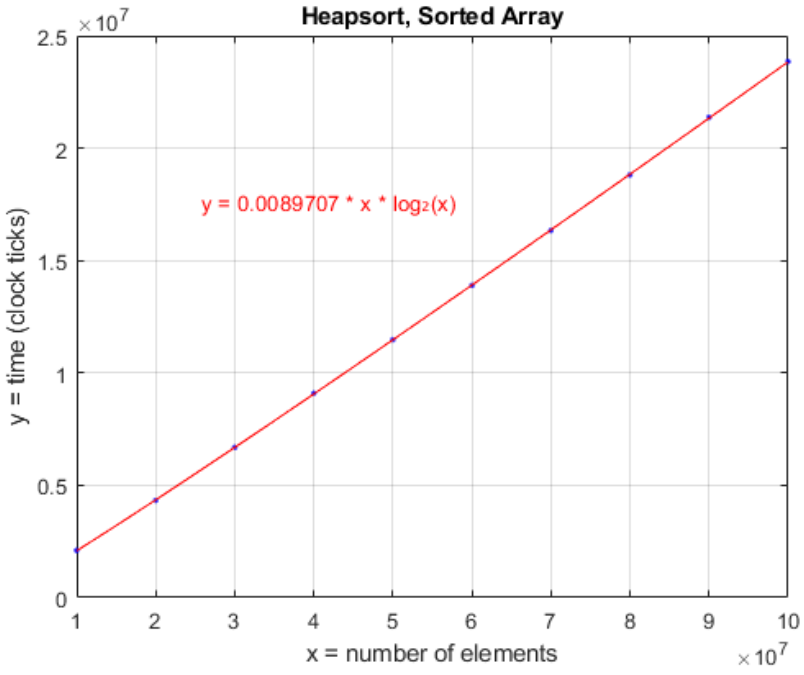


1.2 Heapsort

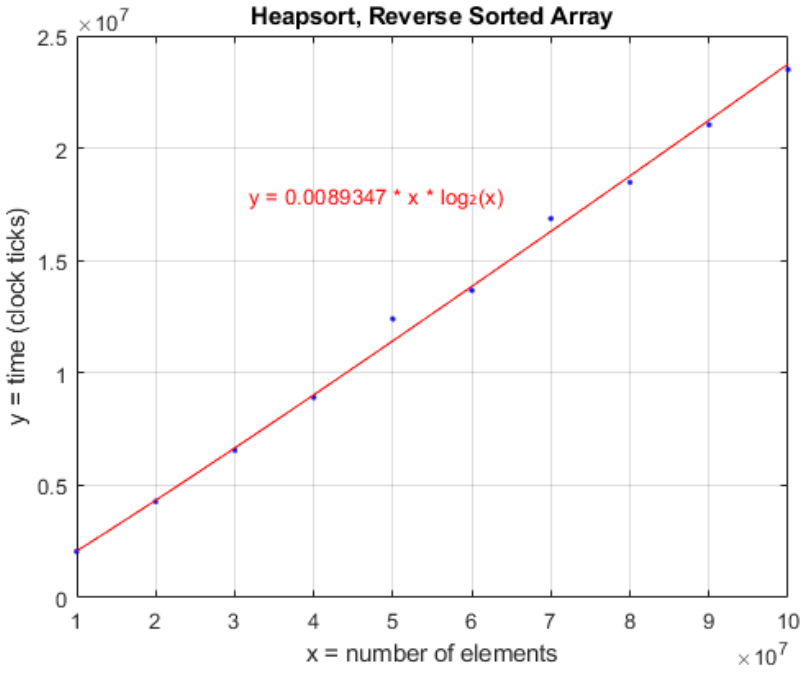
1.2.1 Random



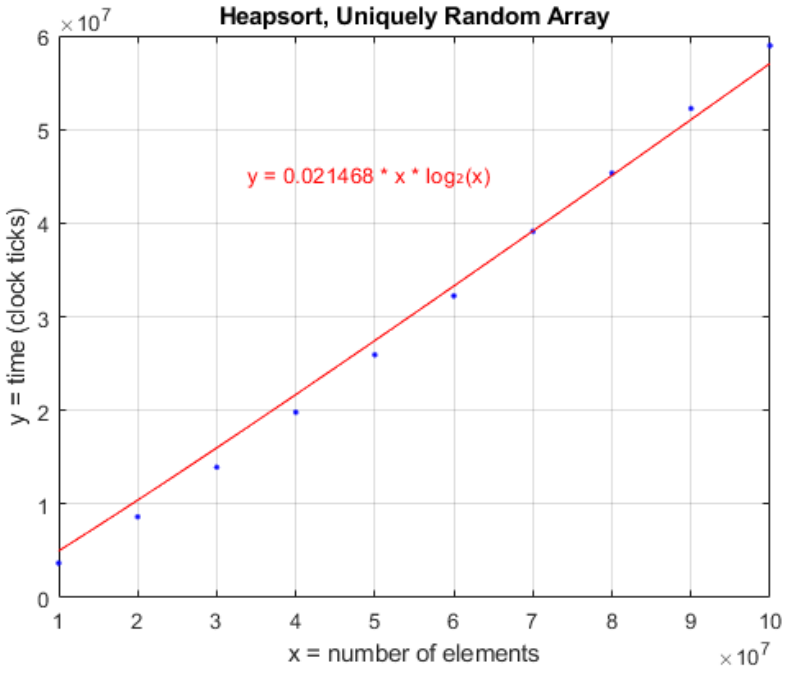
1.2.2 Sorted


Heapsort, Sorted Array
×10^7
2.5
2
1.5
1
0.5
0
y = time (clock ticks)
y = 0.0089707 * x * log2(x)
1
2
3
4
5
6
7
8
9
10
x = number of elements
×10^7

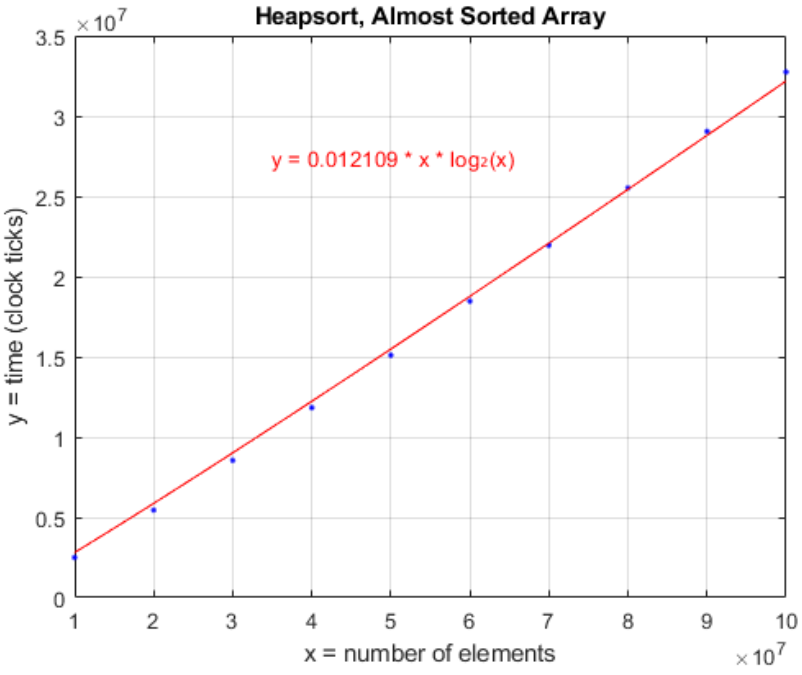
1.2.3 Reverse Sorted



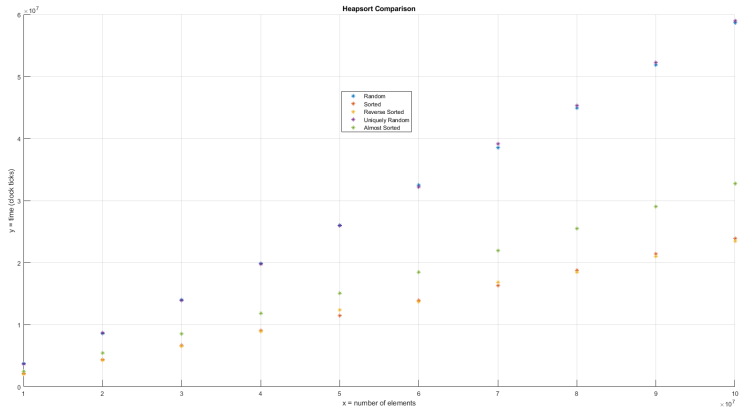
1.2.4 Uniquely Random



1.2.5 Almost Sorted

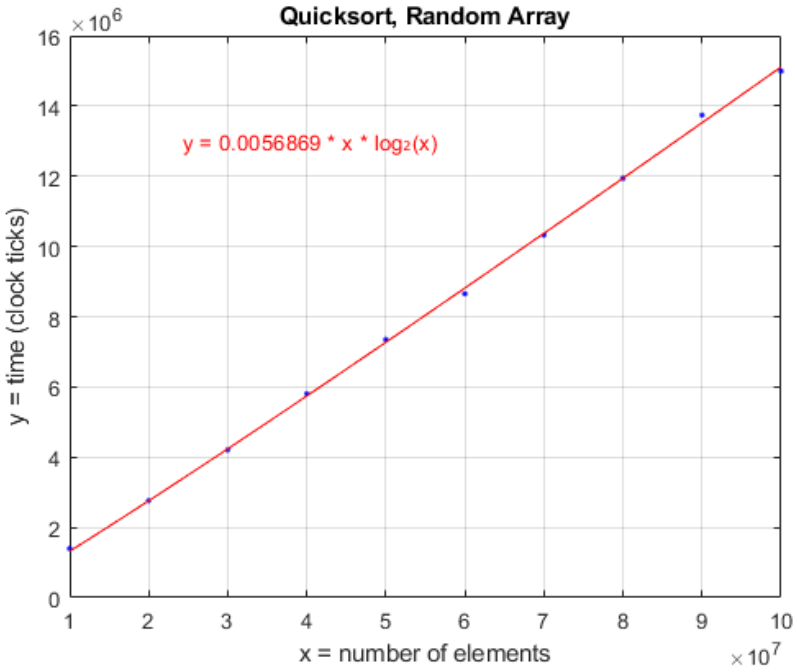


1.2.6 Comparison

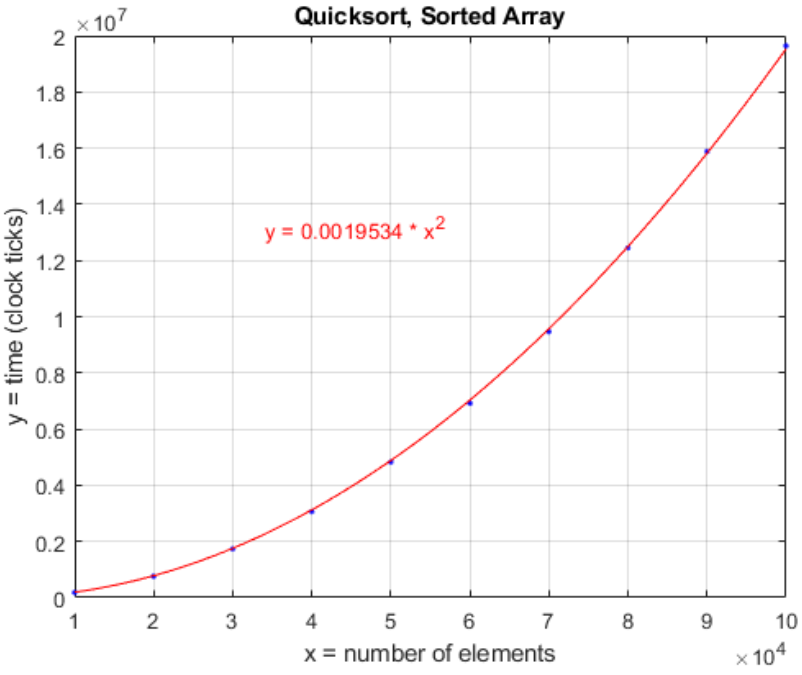


1.3 Quicksort

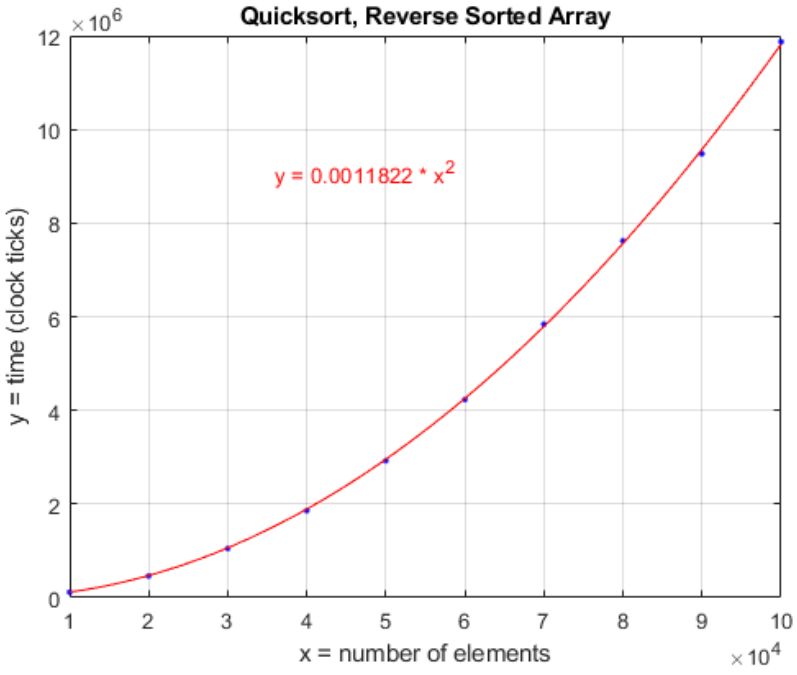
1.3.1 Random



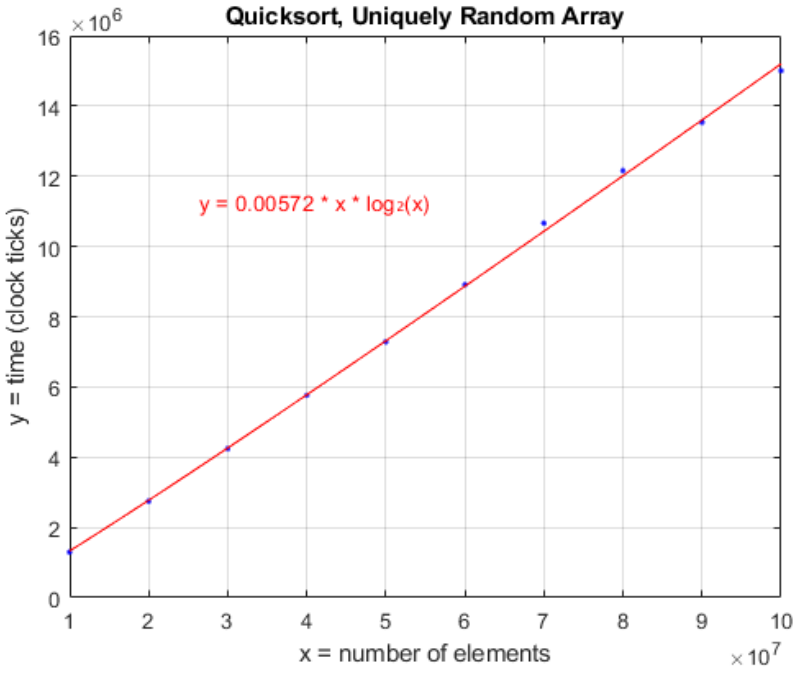
1.3.2 Sorted


Quicksort, Sorted Array
×10^7
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
y = time (clock ticks)
y = 0.0019534 * x^2
1
2
3
4
5
6
7
8
9
10
x = number of elements
×10^4

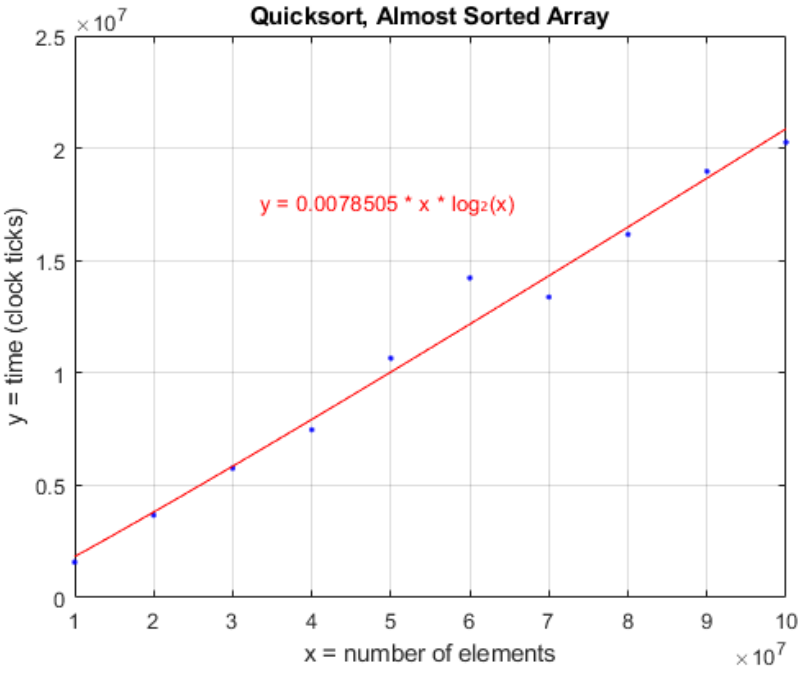
1.3.3 Reverse Sorted



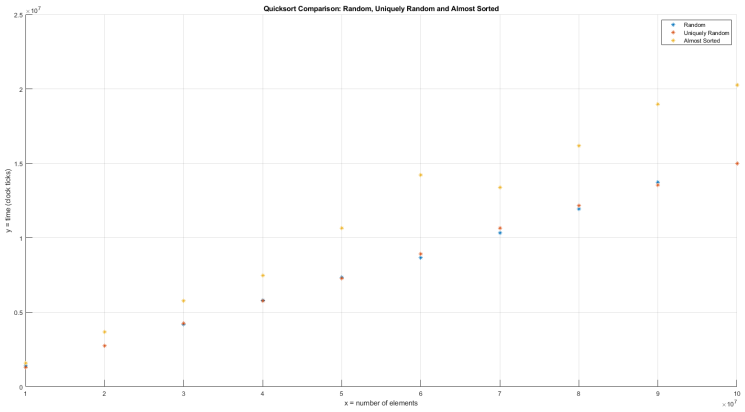
1.3.4 Uniquely Random



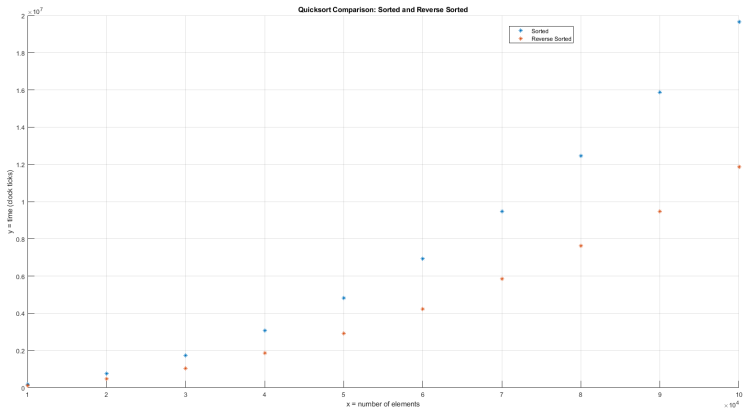
1.3.5 Almost Sorted



1.3.6 Comparison - Random, Uniquely Random and Almost Sorted

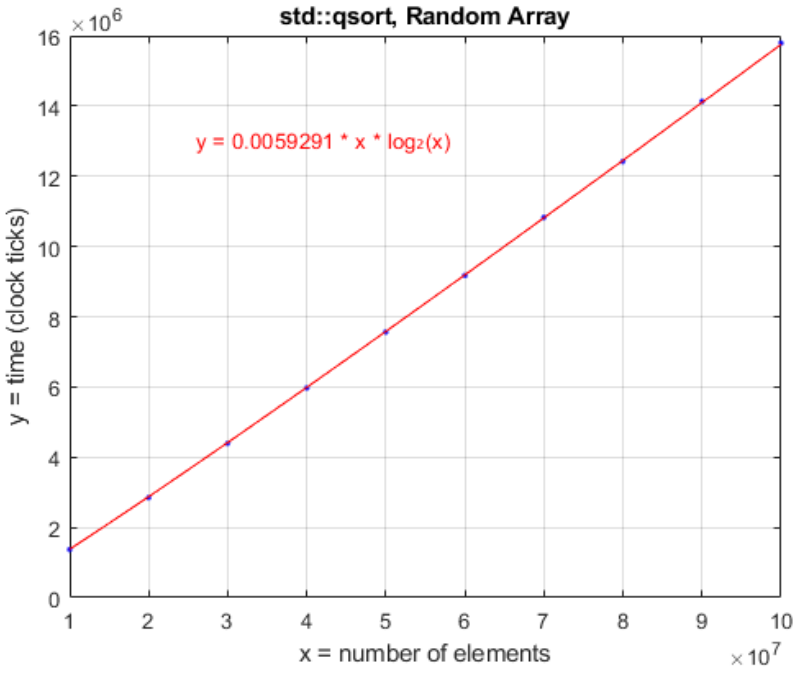


1.3.7 Comparison - Sorted and Reverse Sorted

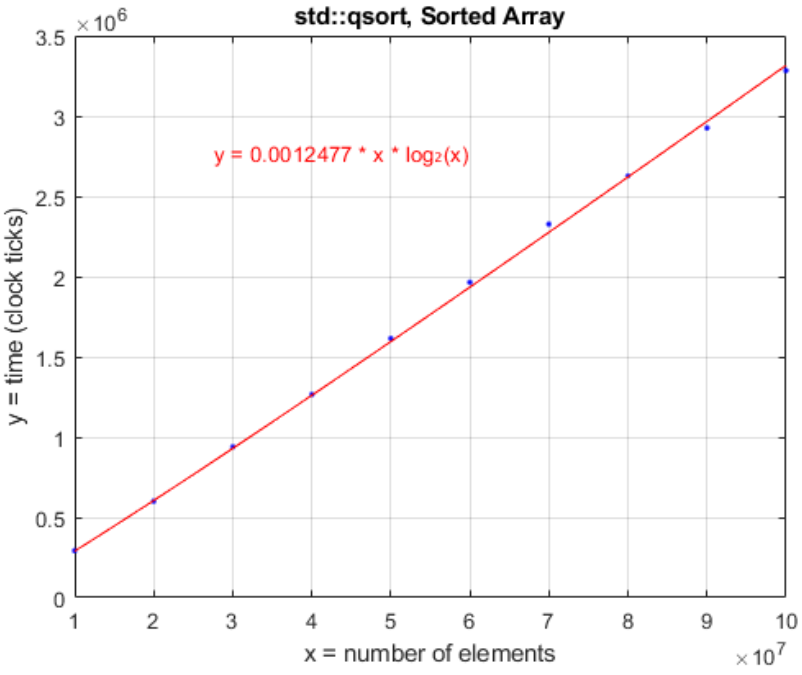


1.4 std::qsort

1.4.1 Random

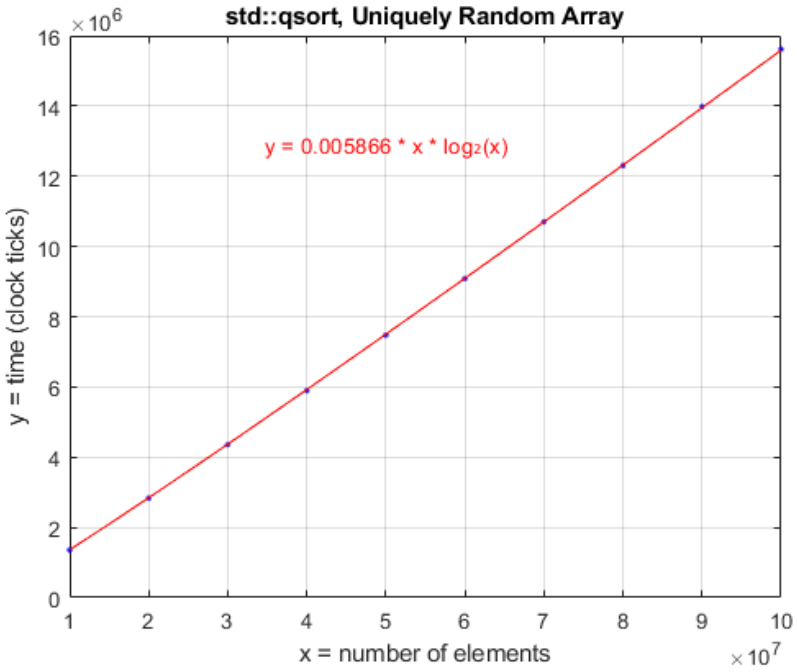


1.4.2 Sorted

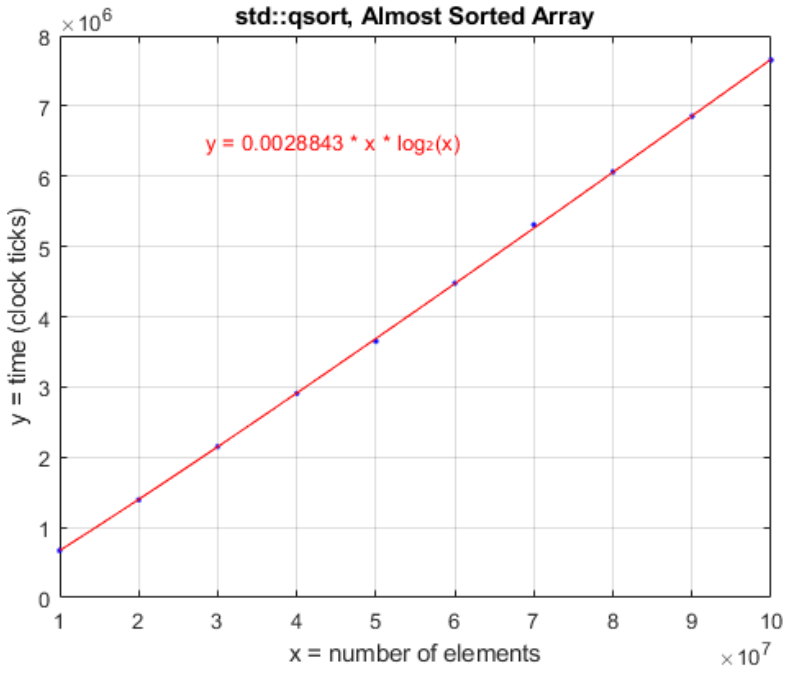

std::qsort, Sorted Array
×10^6
3.5
3
2.5
2
1.5
1
0.5
0
y = time (clock ticks)
y = 0.0012477 * x * log2(x)
1
2
3
4
5
6
7
8
9
10
x = number of elements
×10^7

1.4.3 Reverse Sorted

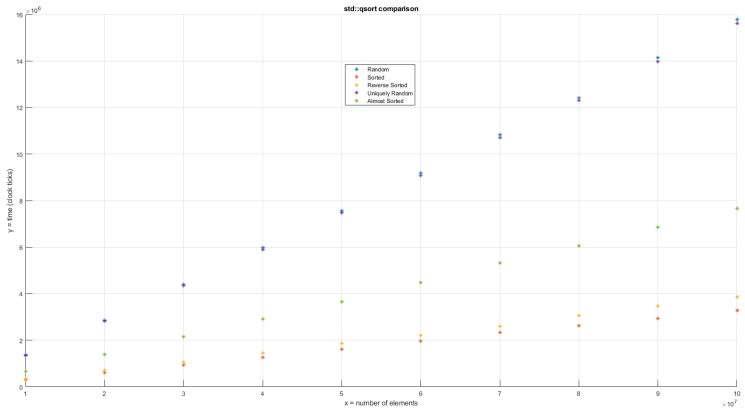
1.4.4 Uniquely Random



1.4.5 Almost Sorted

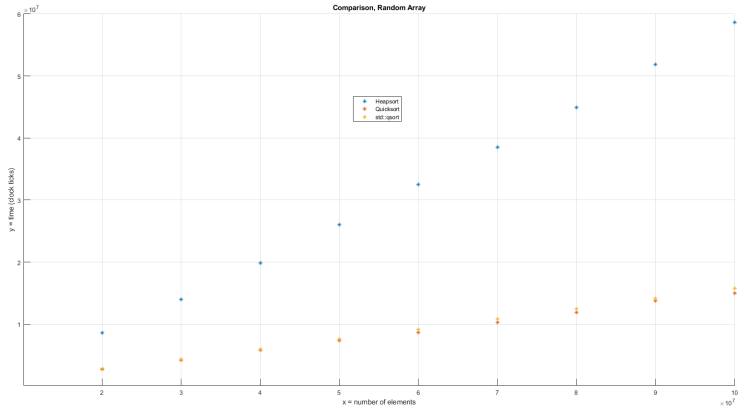


1.4.6 Comparison

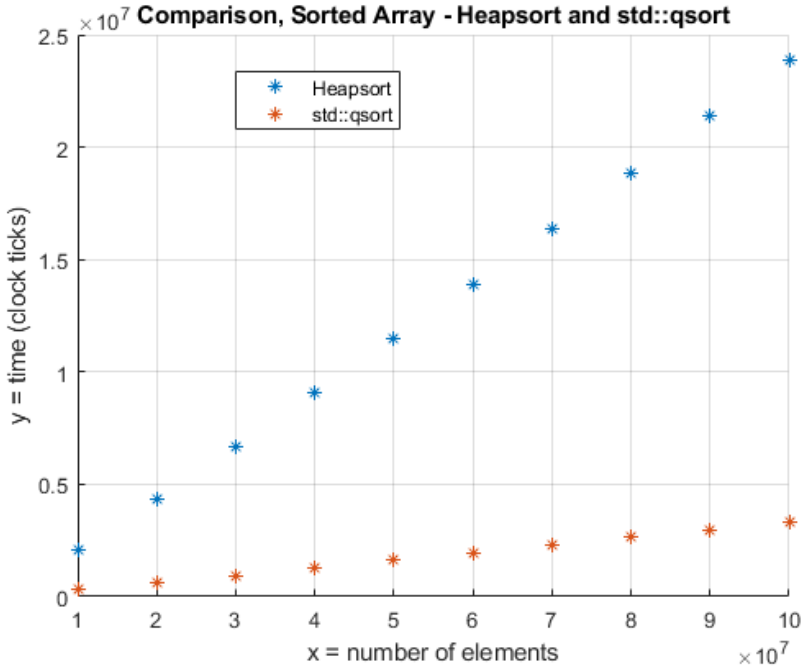


1.5 Comparison

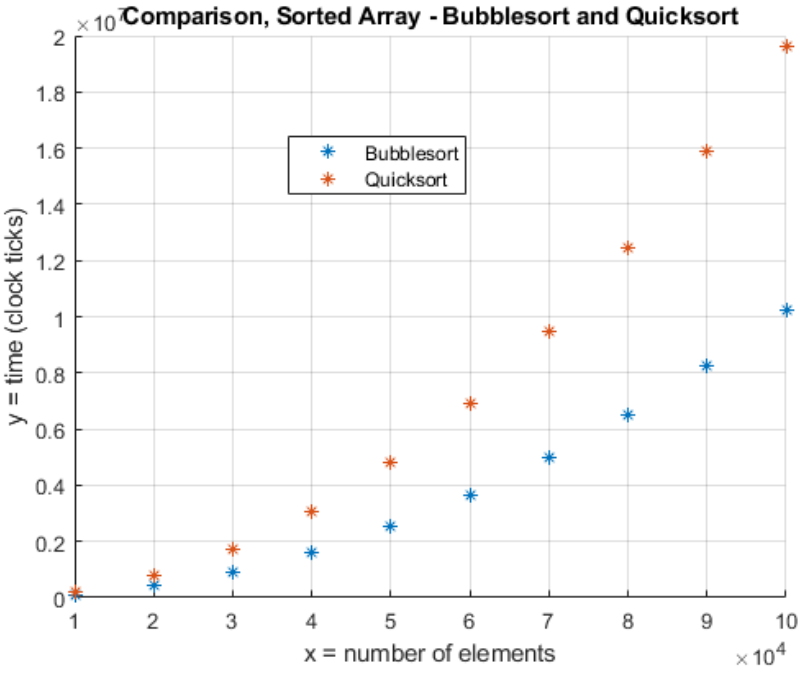
1.5.1 Random



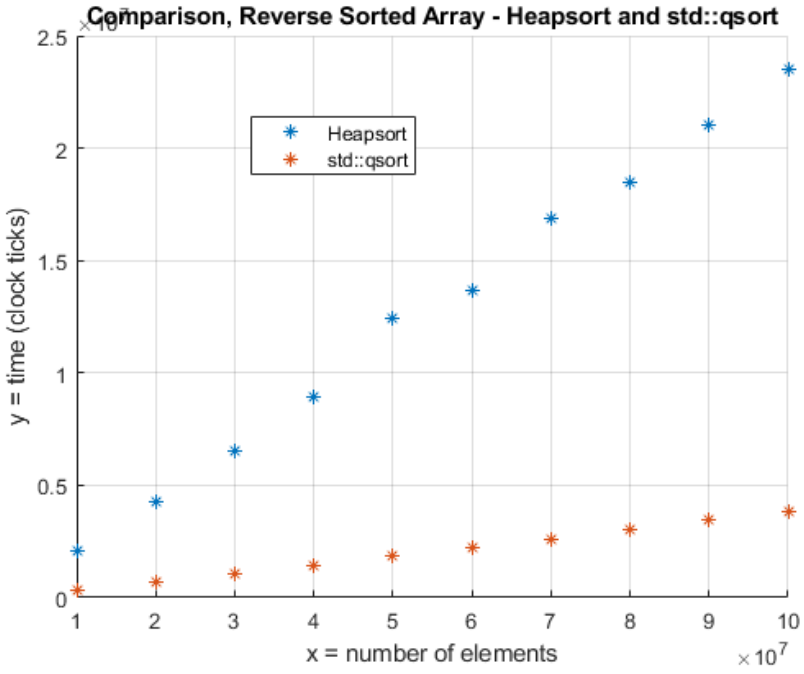
1.5.2 Sorted - Heapsort and std::qsort



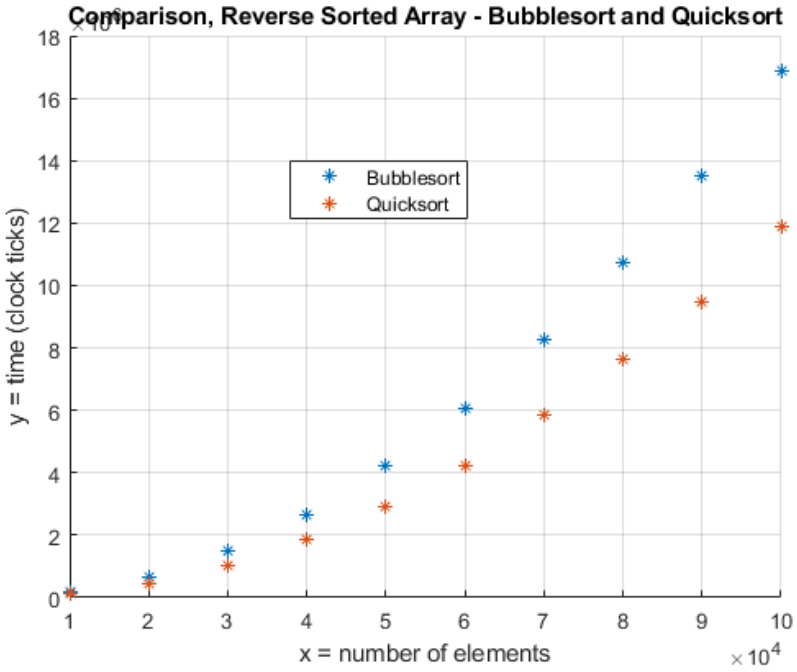
1.5.3 Sorted - Bubblesort and Quicksort



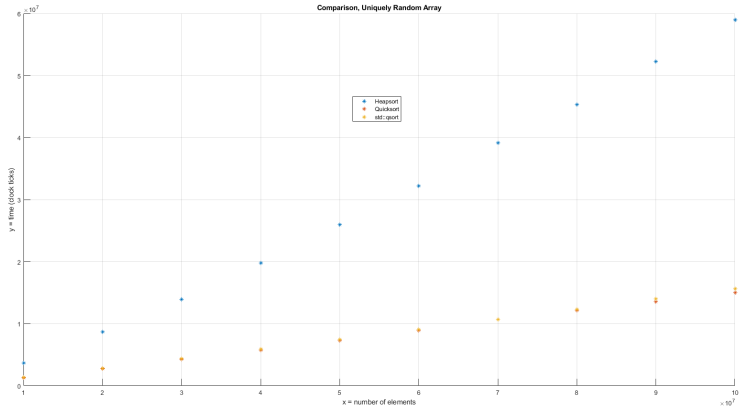
1.5.4 Reverse Sorted - Heapsort and std::qsort



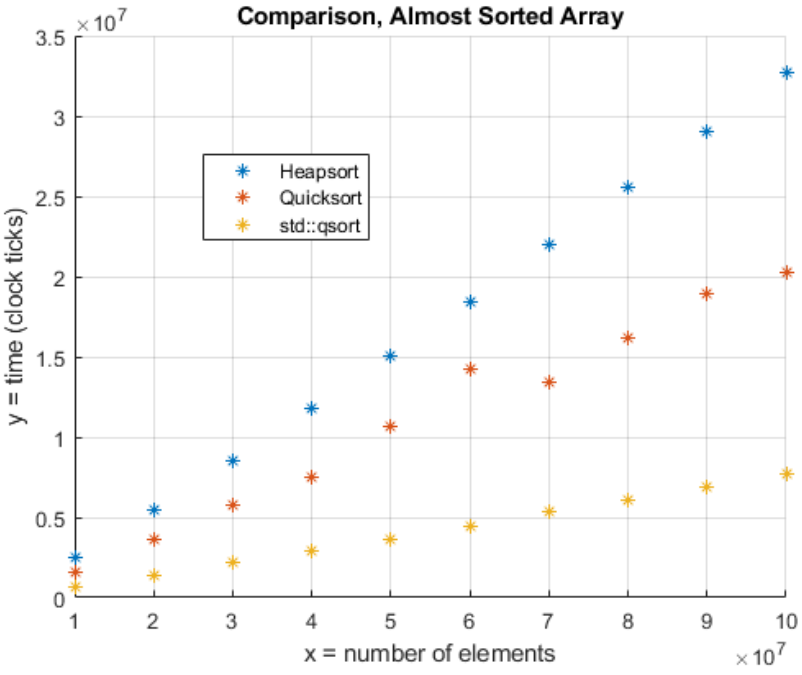
1.5.5 Reverse Sorted - Bubblesort and Quicksort



1.5.6 Uniquely Random



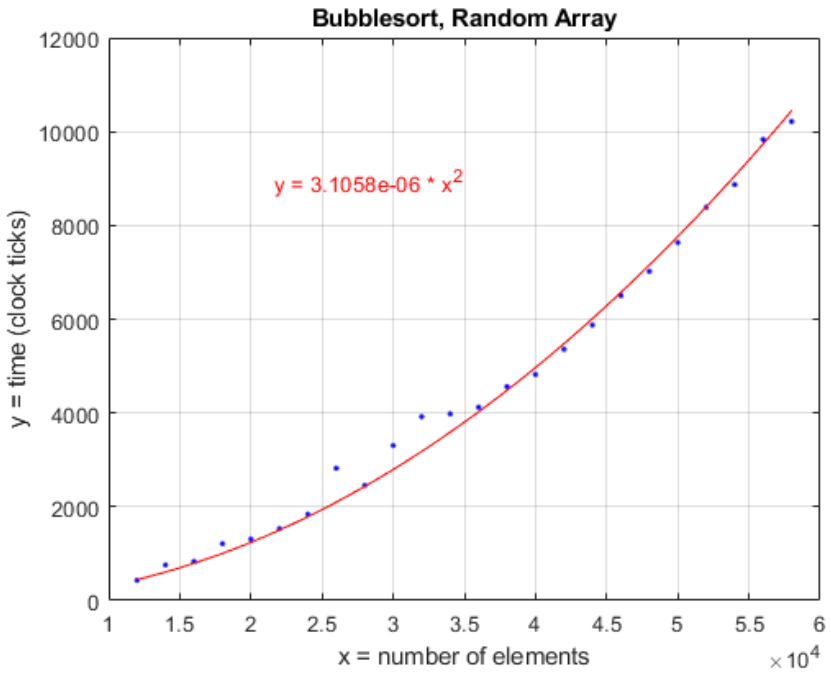
1.5.7 Almost Sorted



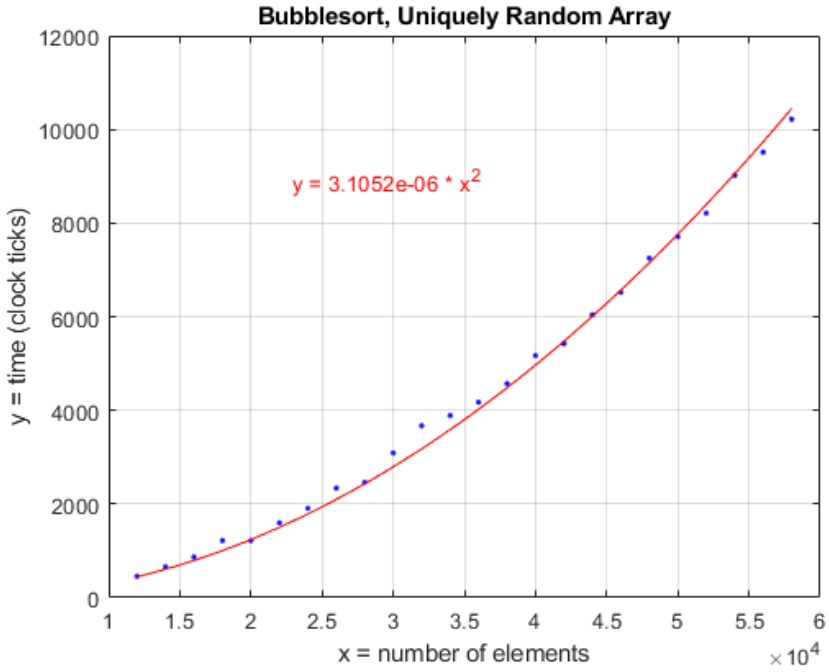
2 Oscar's tests 1

2.1 Bubblesort

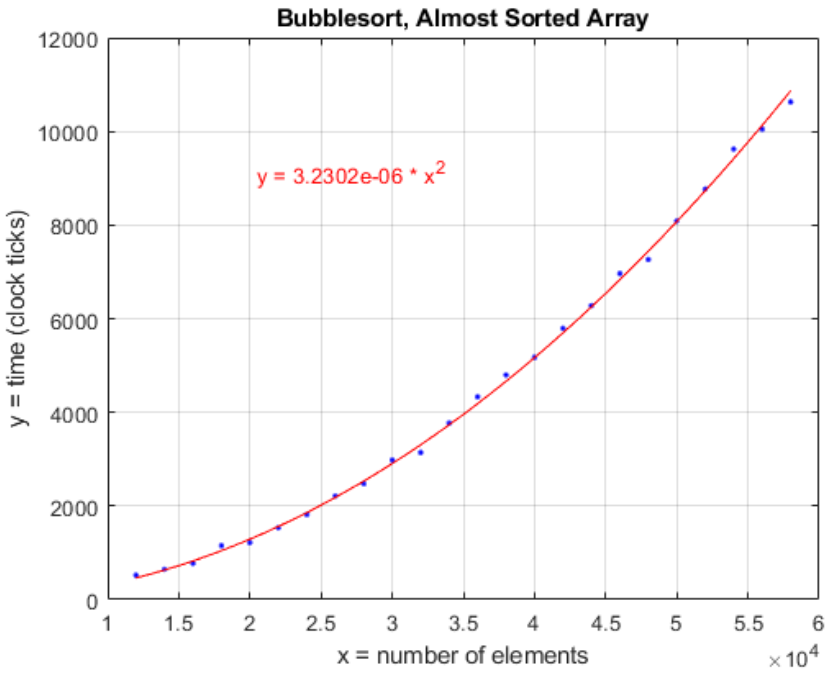
2.1.1 Random



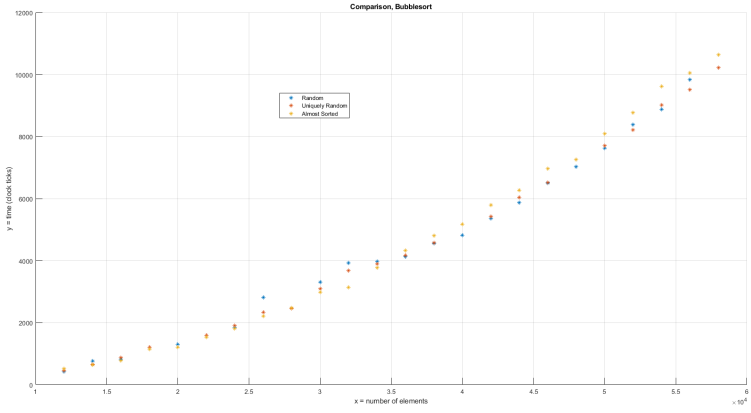
2.1.2 Uniquely Random



2.1.3 Almost Sorted

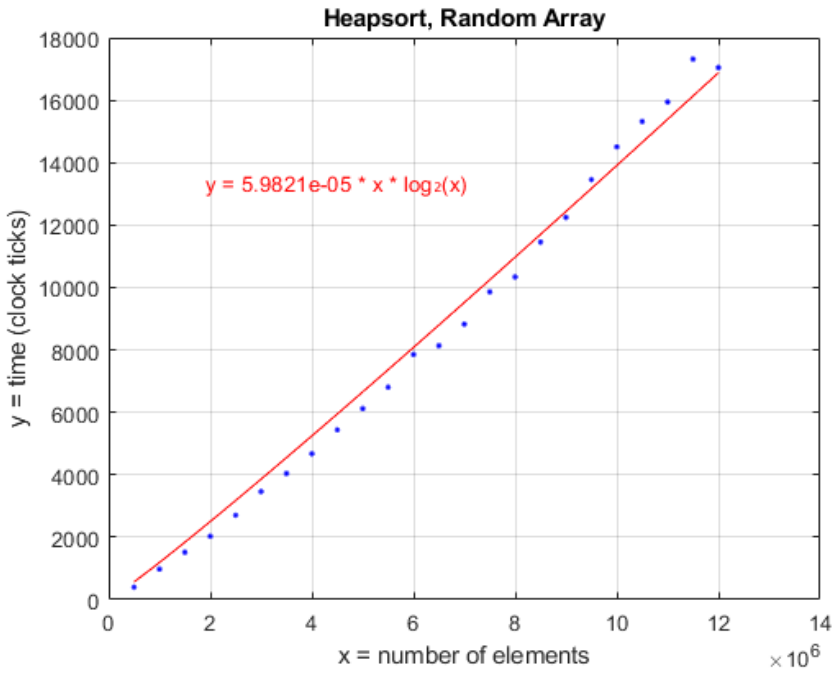


2.1.4 Comparison

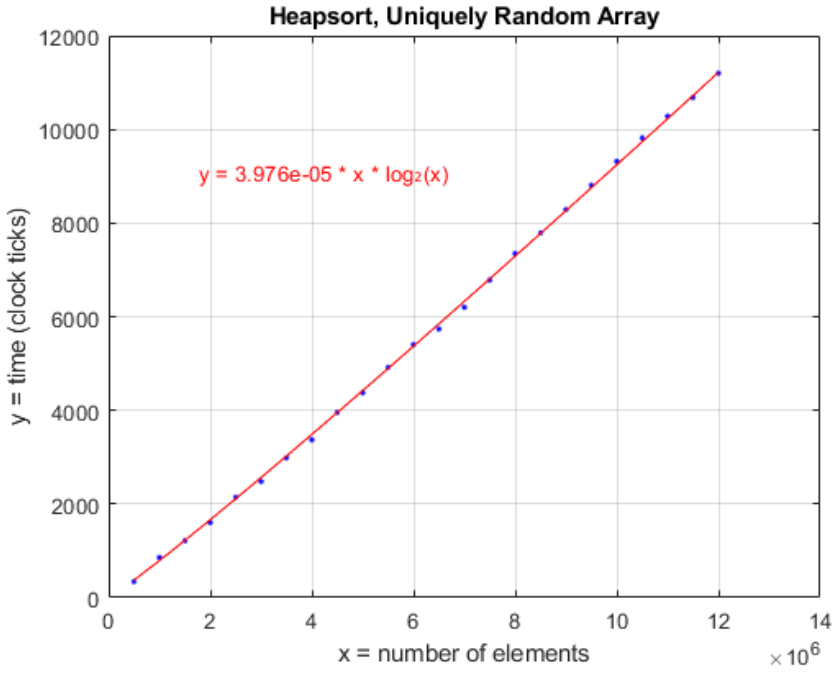


2.2 Heapsort

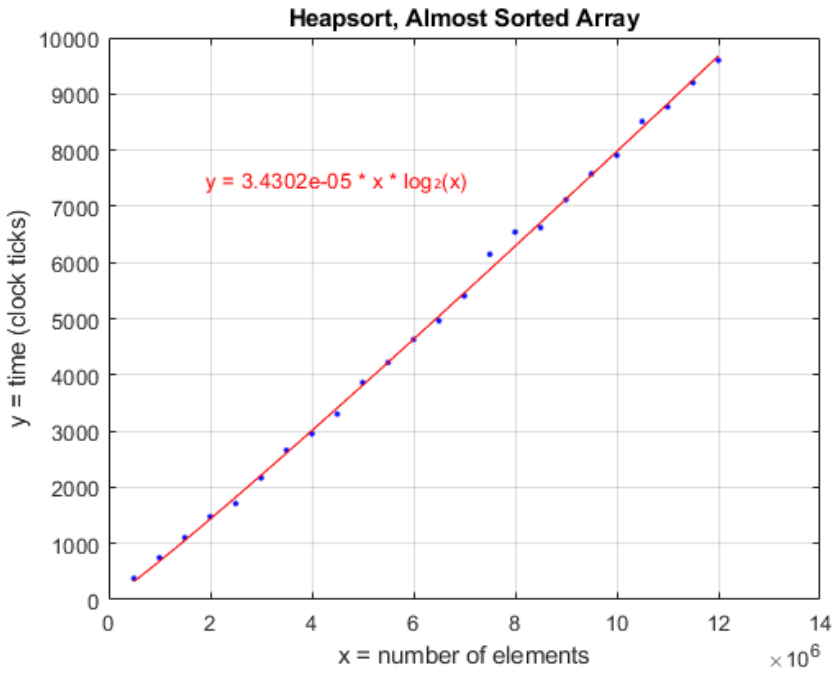
2.2.1 Random



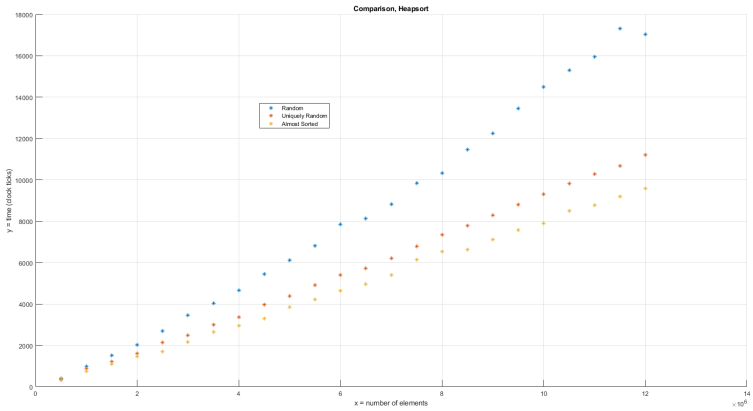
2.2.2 Uniquely Random



2.2.3 Almost Sorted

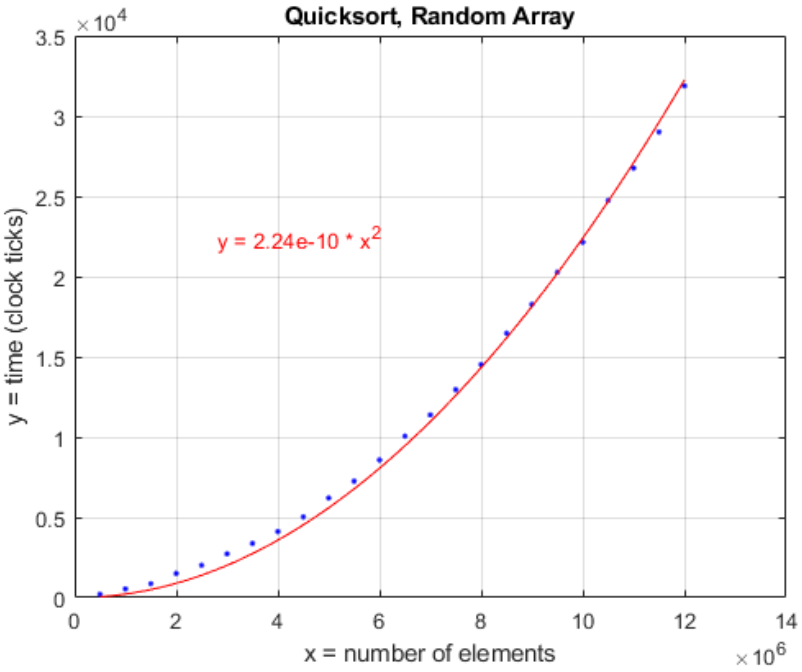


2.2.4 Comparison

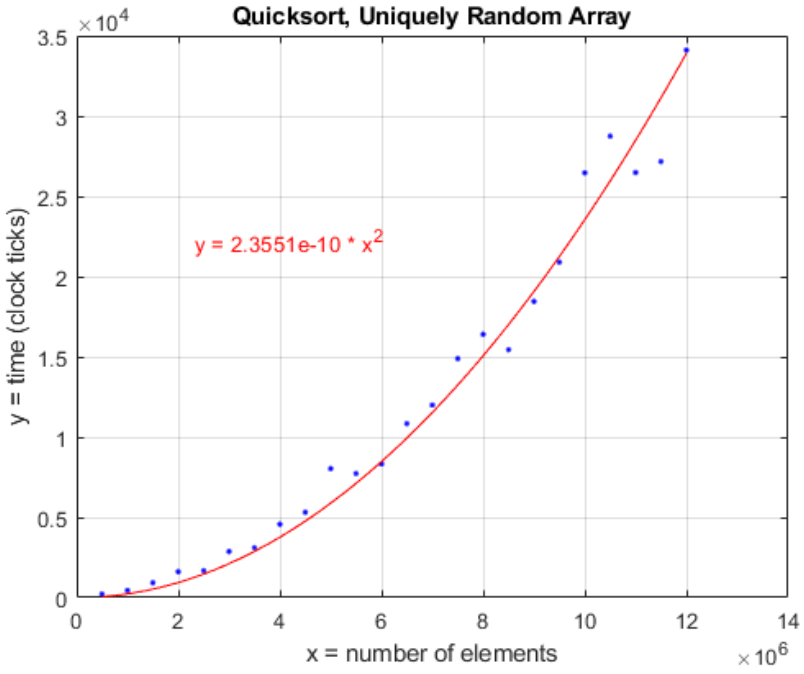


2.3 Quicksort

2.3.1 Random

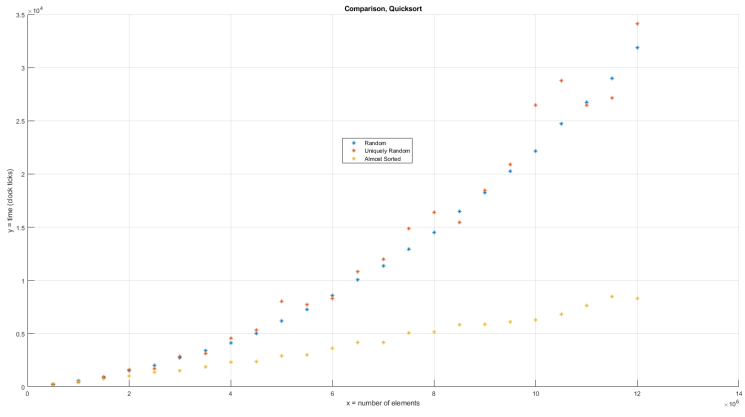


2.3.2 Uniquely Random



2.3.3 Almost Sorted

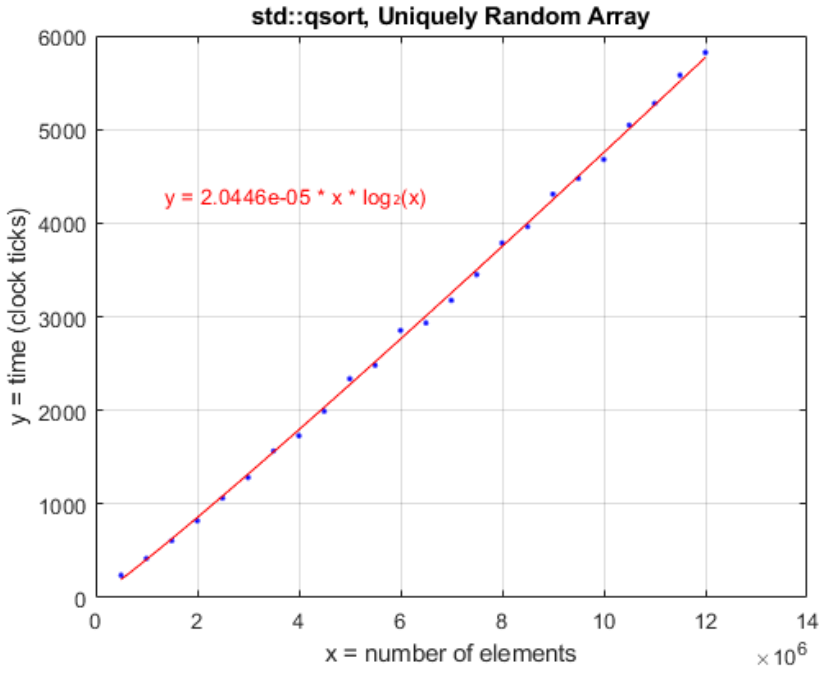
2.3.4 Comparison



2.4 std::qsort

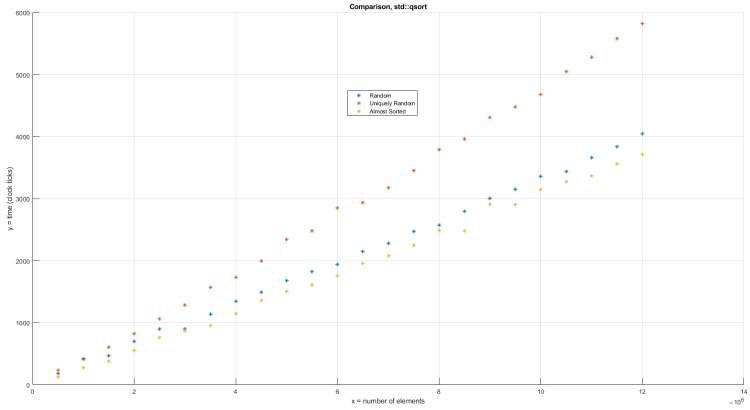
2.4.1 Random

2.4.2 Uniquely Random



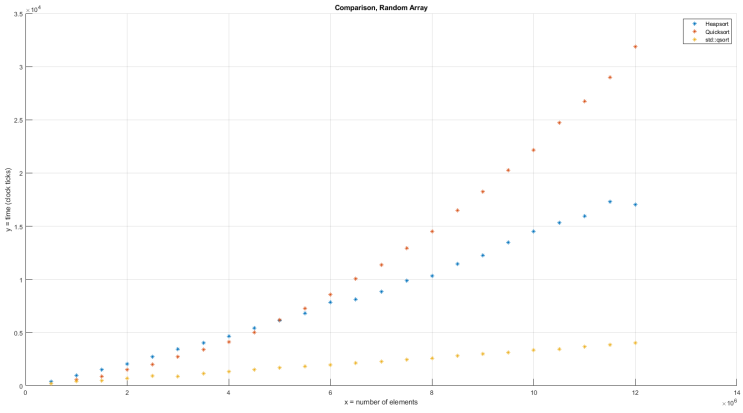
2.4.3 Almost Sorted

2.4.4 Comparison

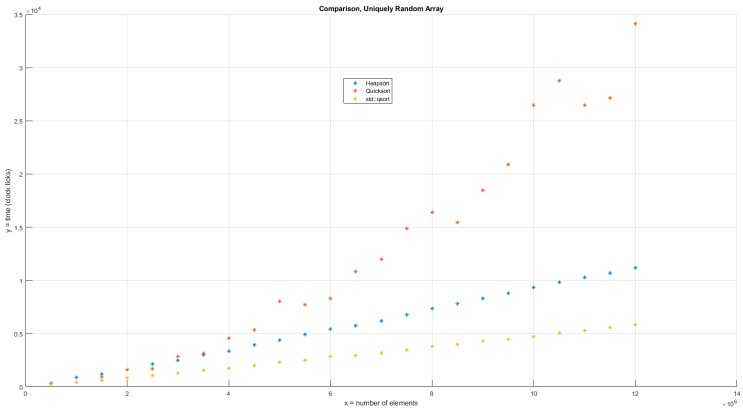


2.5 Comparison

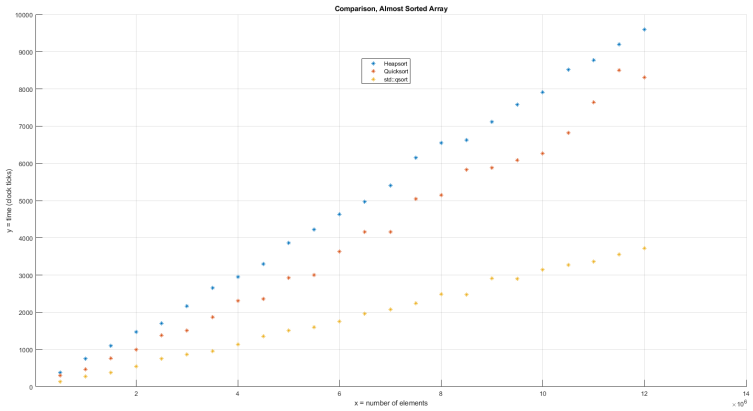
2.5.1 Random



2.5.2 Uniquely Random



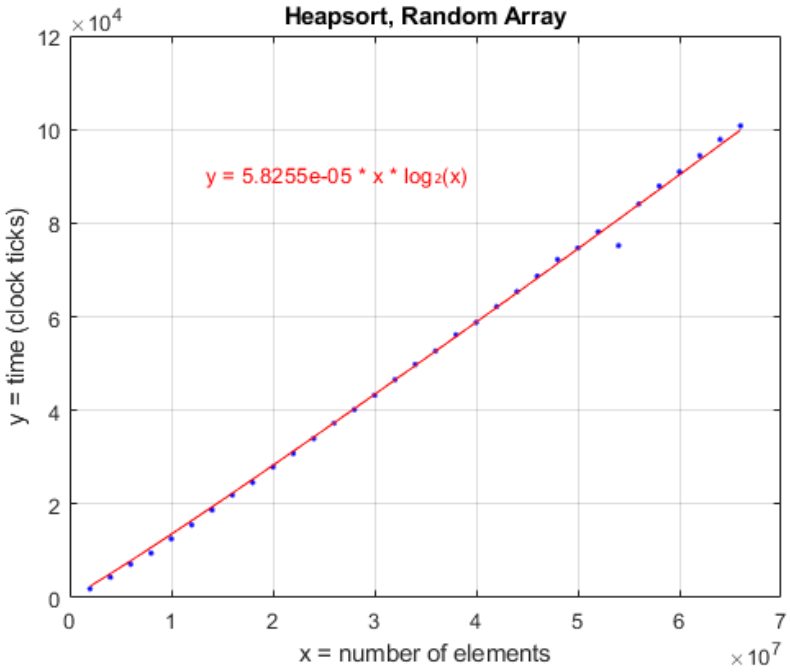
2.5.3 Almost Sorted



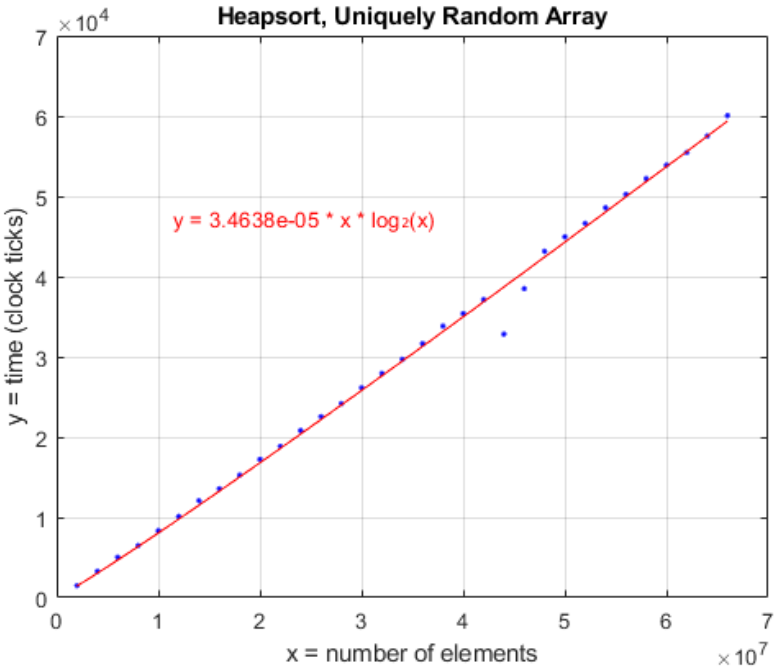
3 Oscar's tests 2

3.1 Heapsort

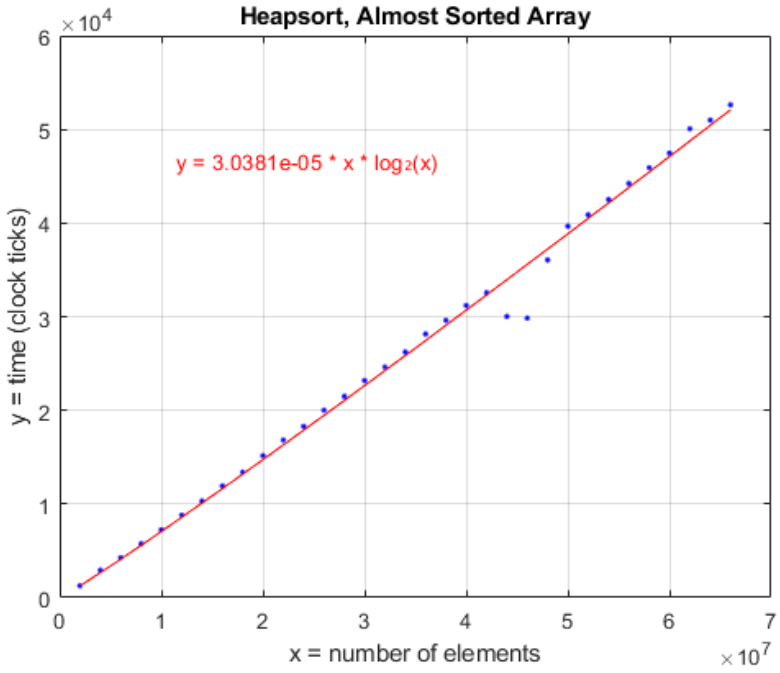
3.1.1 Random



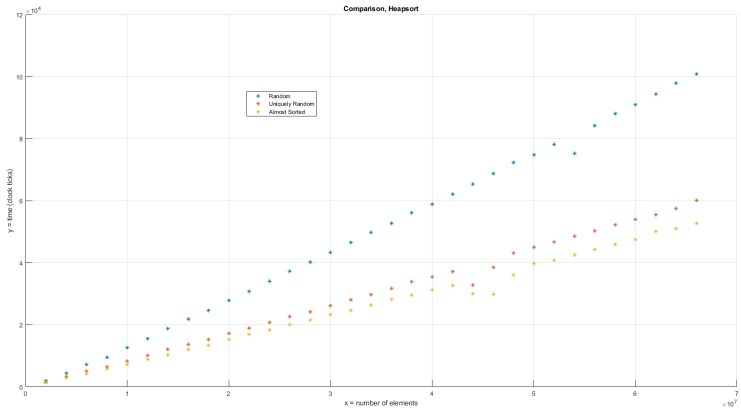
3.1.2 Uniquely Random



3.1.3 Almost Sorted

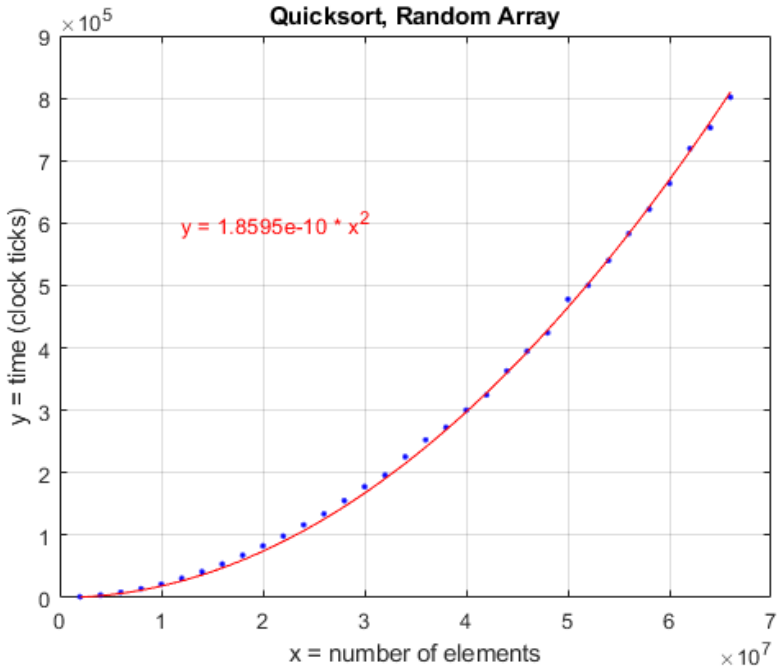


3.1.4 Comparison

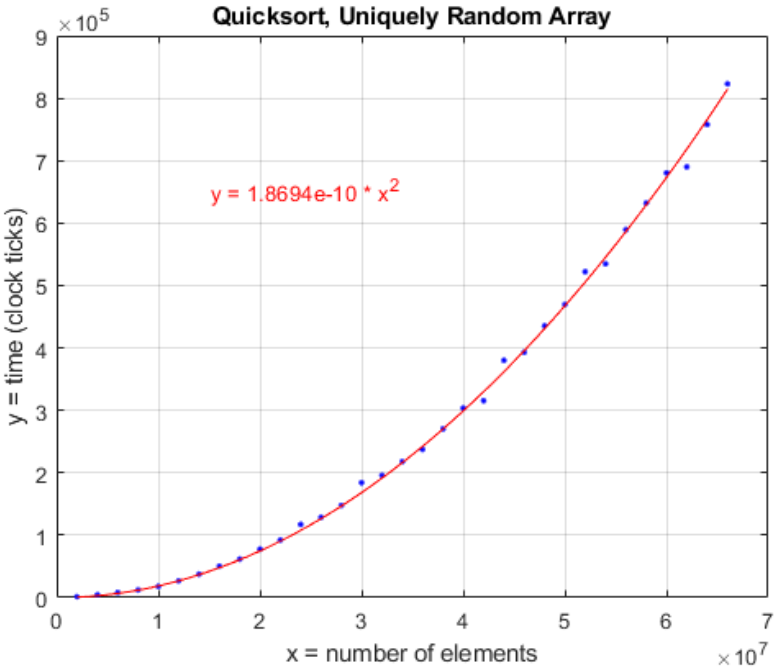


3.2 Quicksort

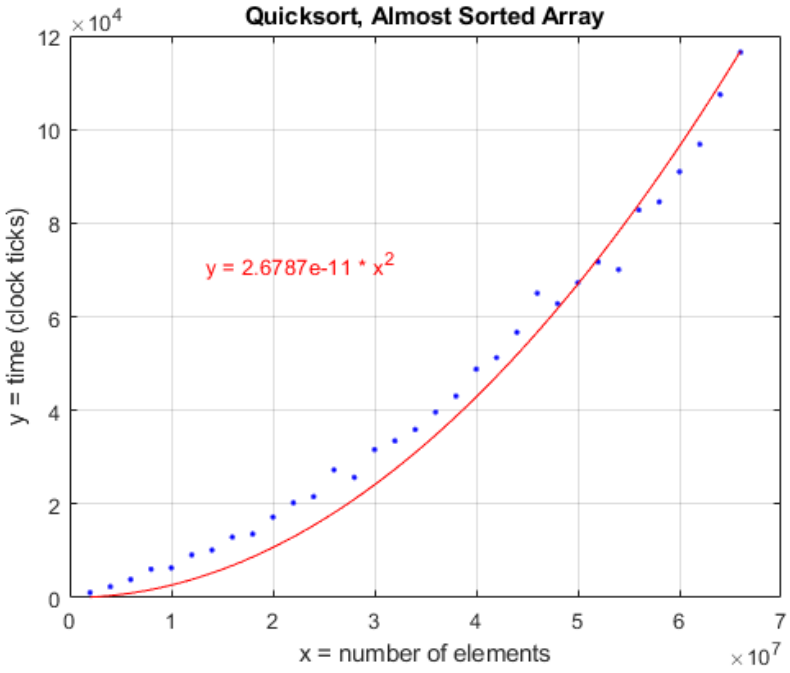
3.2.1 Random



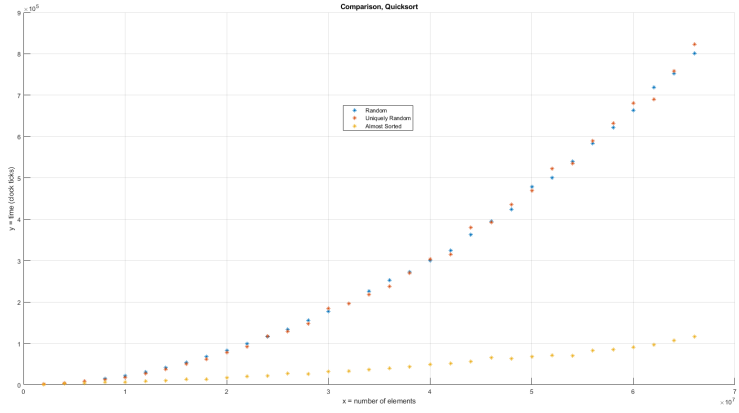
3.2.2 Uniquely Random



3.2.3 Almost Sorted

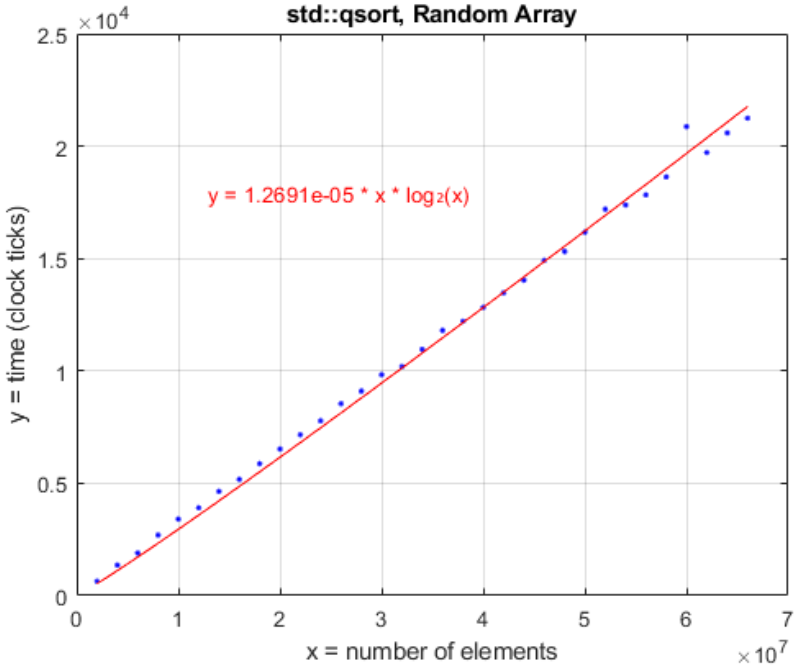


3.2.4 Comparison

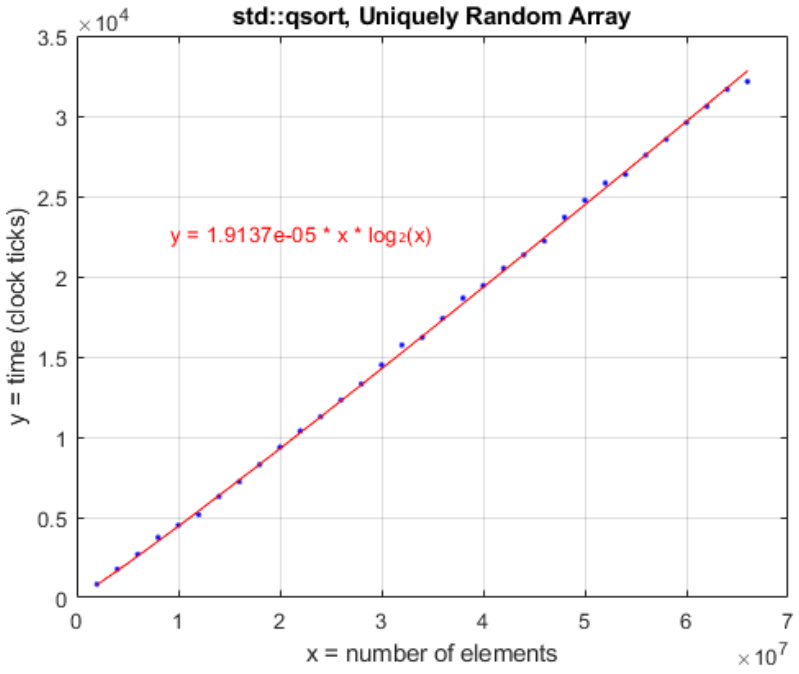


3.3 std::qsort

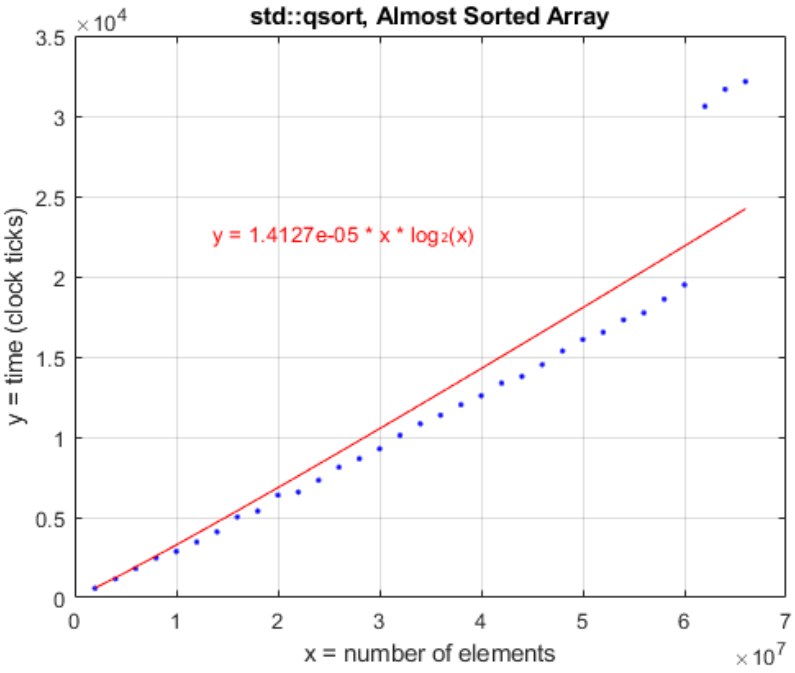
3.3.1 Random



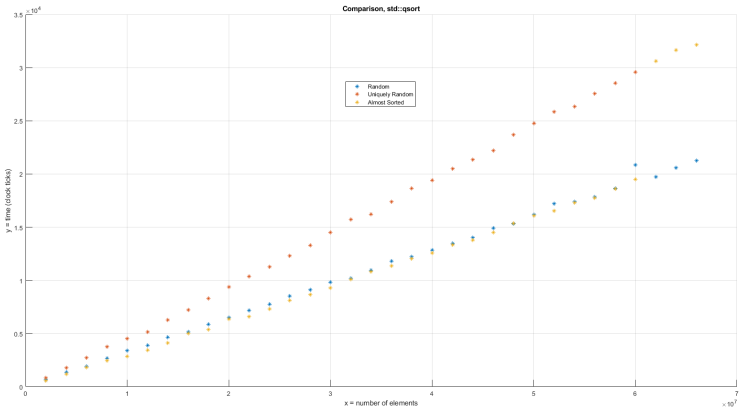
3.3.2 Uniquely Random



3.3.3 Almost Sorted

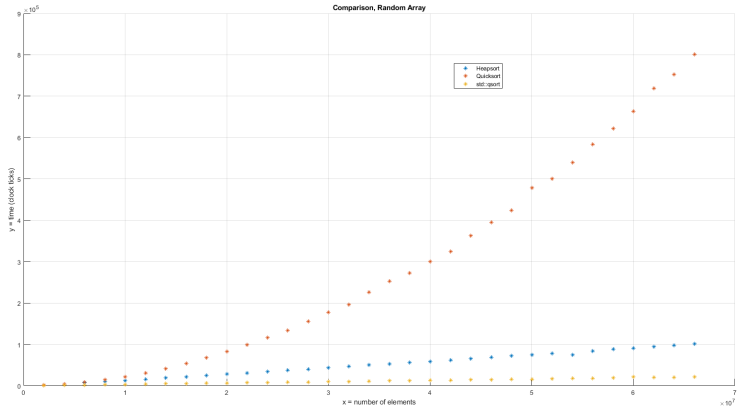


3.3.4 Comparison

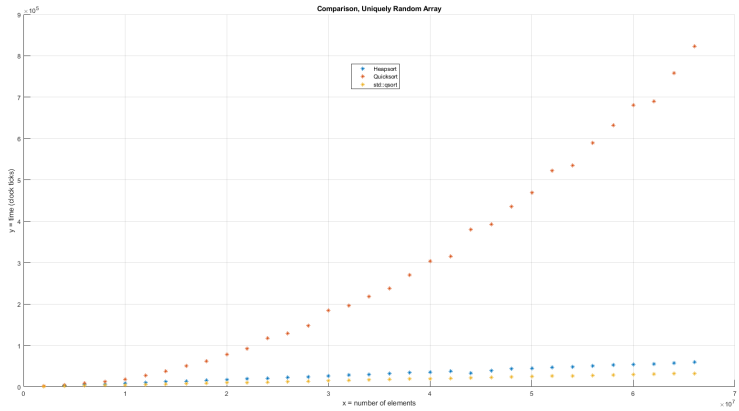
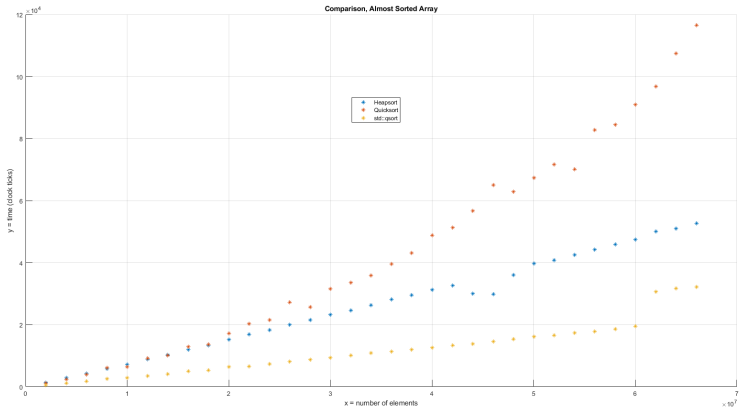


3.4 Comparison

3.4.1 Random



3.4.2 Uniquely Random



3.4.3 Almost Sorted