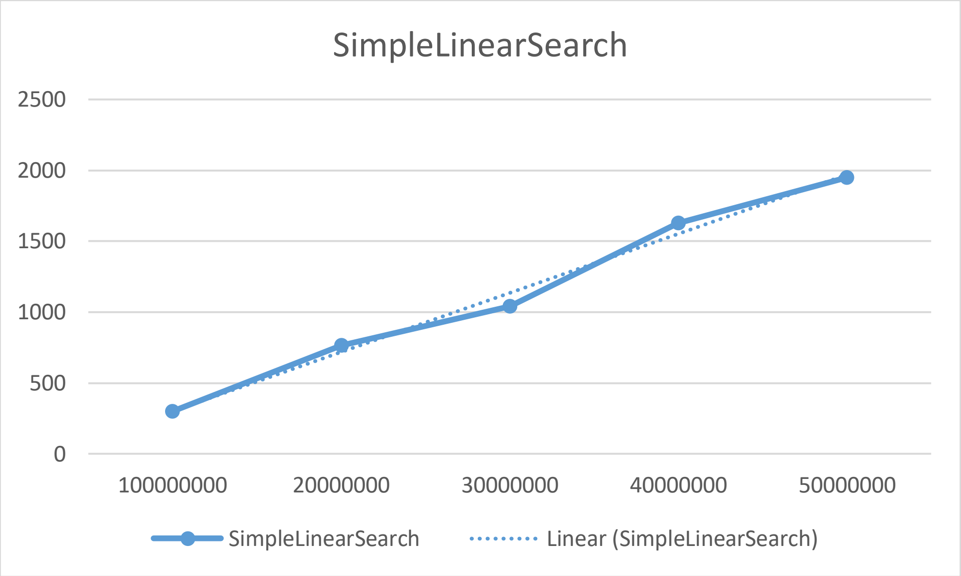
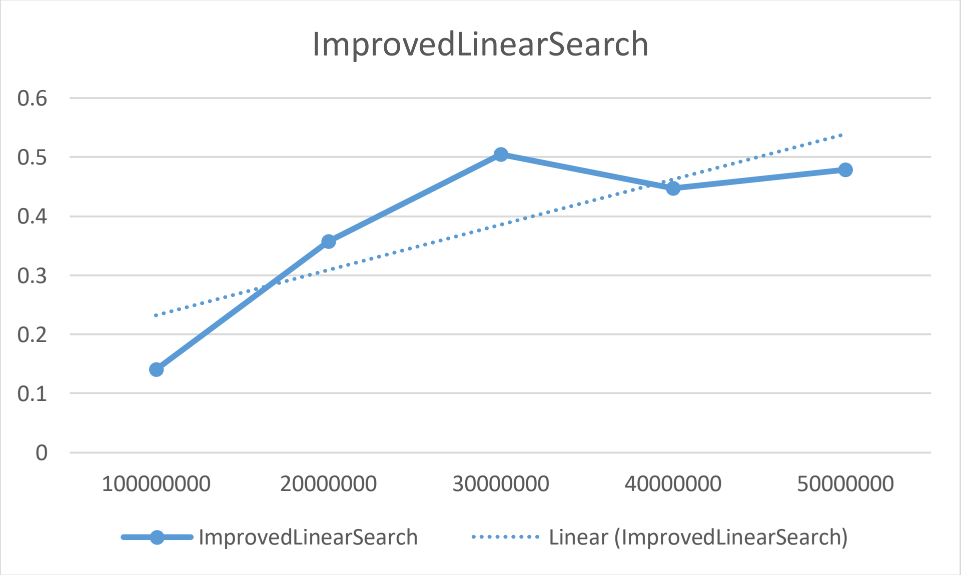
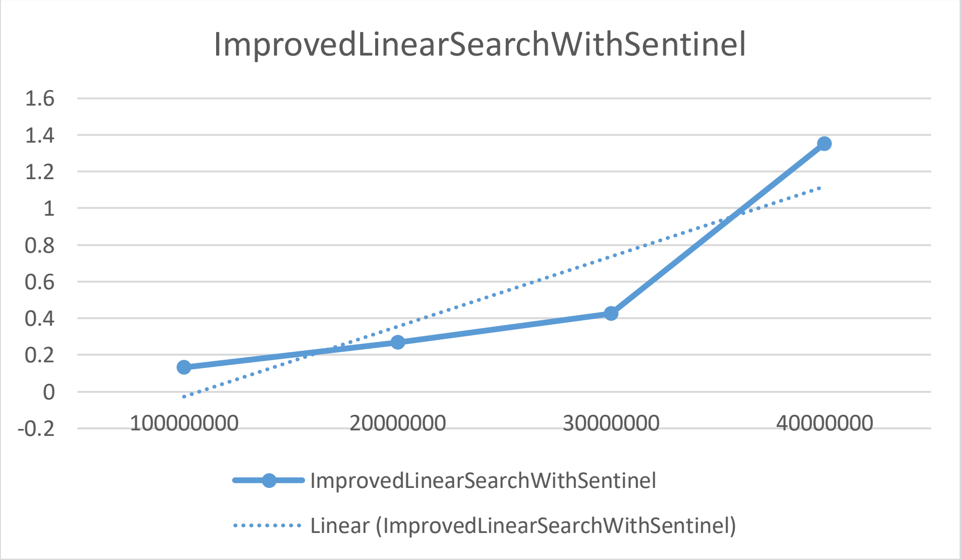
Serhii Holishevskyi

Note: OY – always time
OX – Time or array size

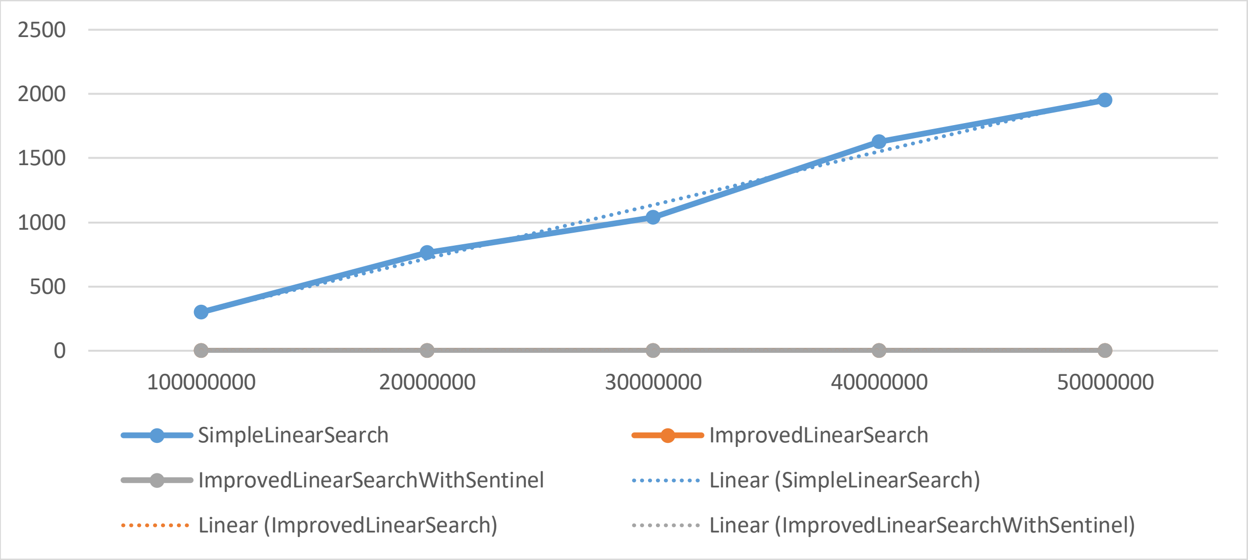
Time(ms) vs array size:



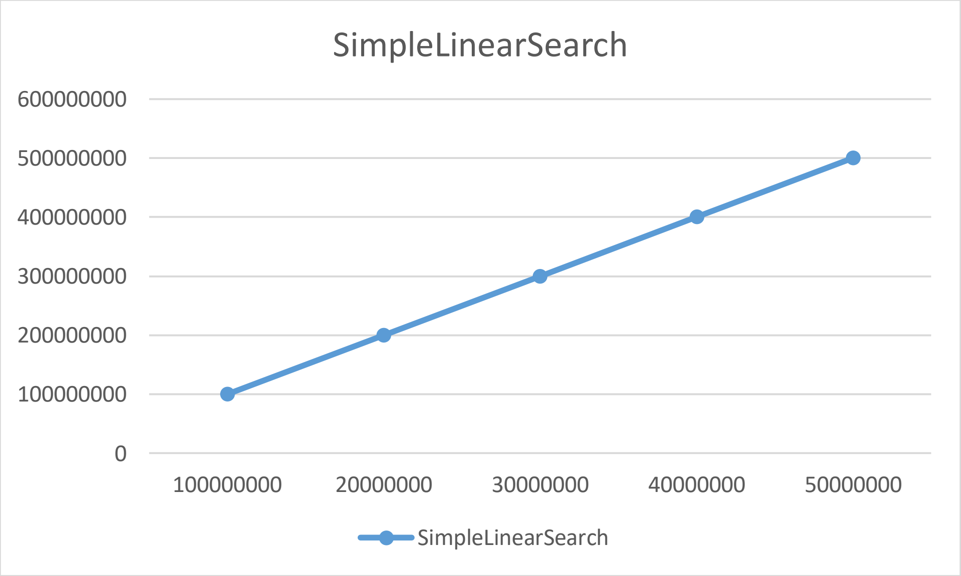


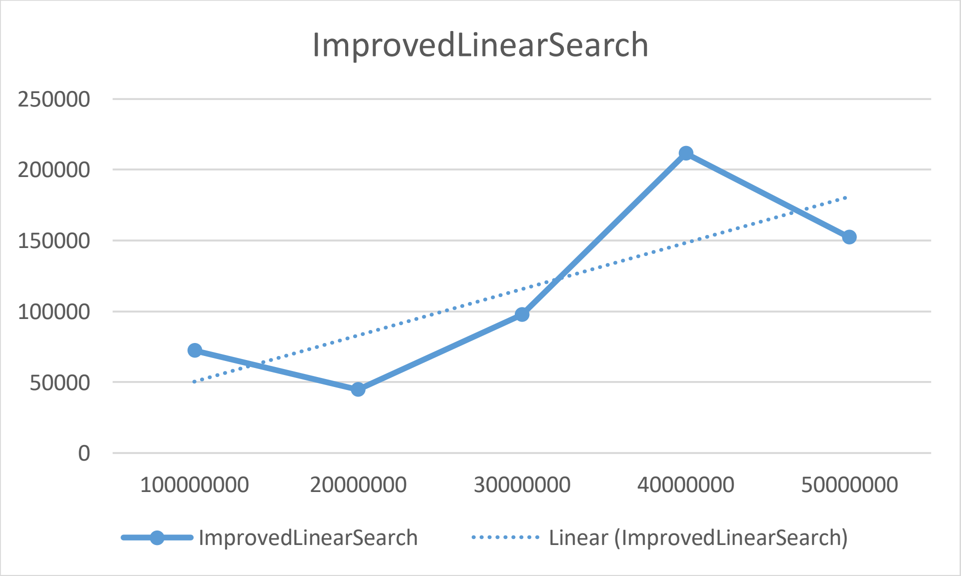


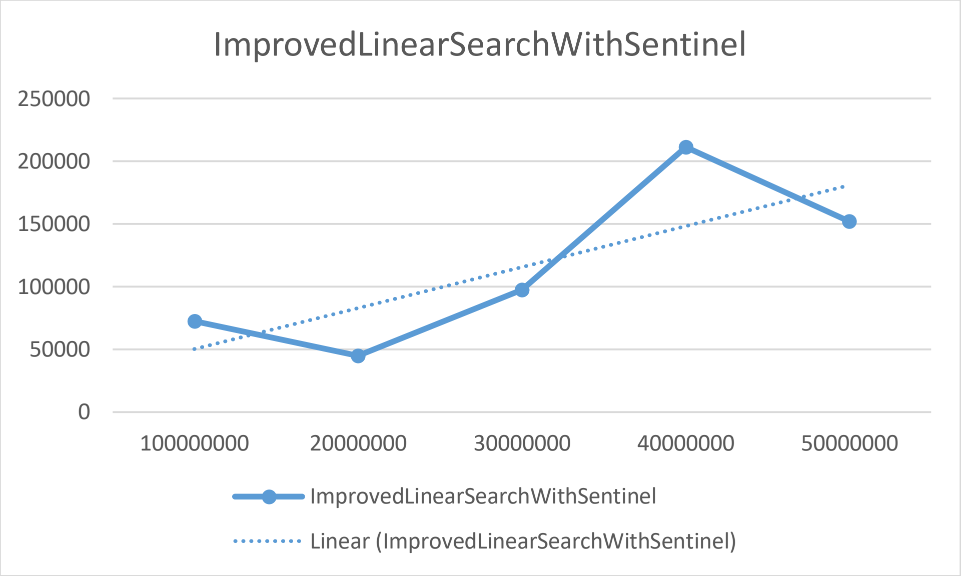
Combined:



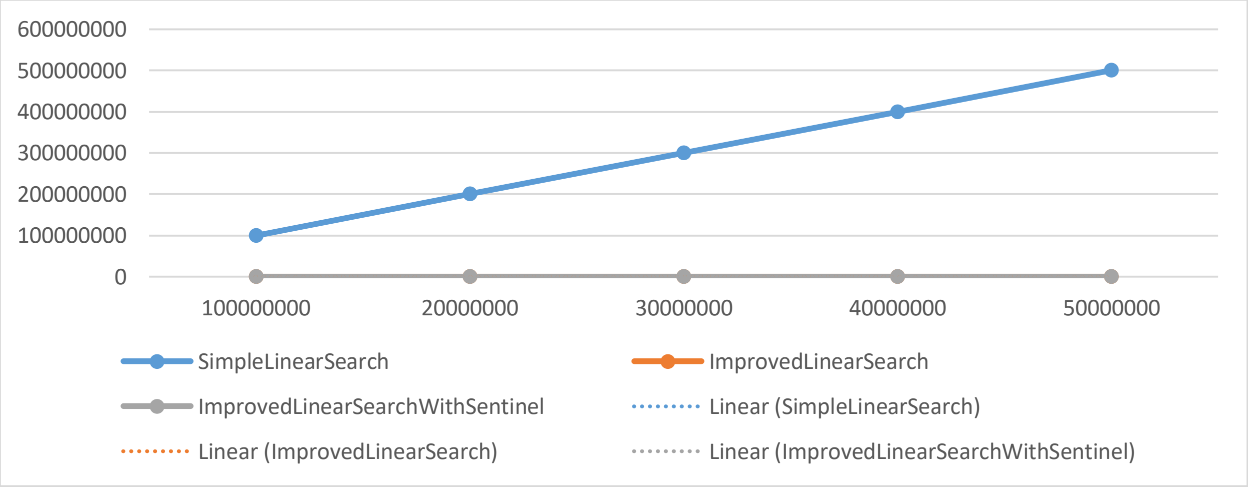
Operations vs array size:





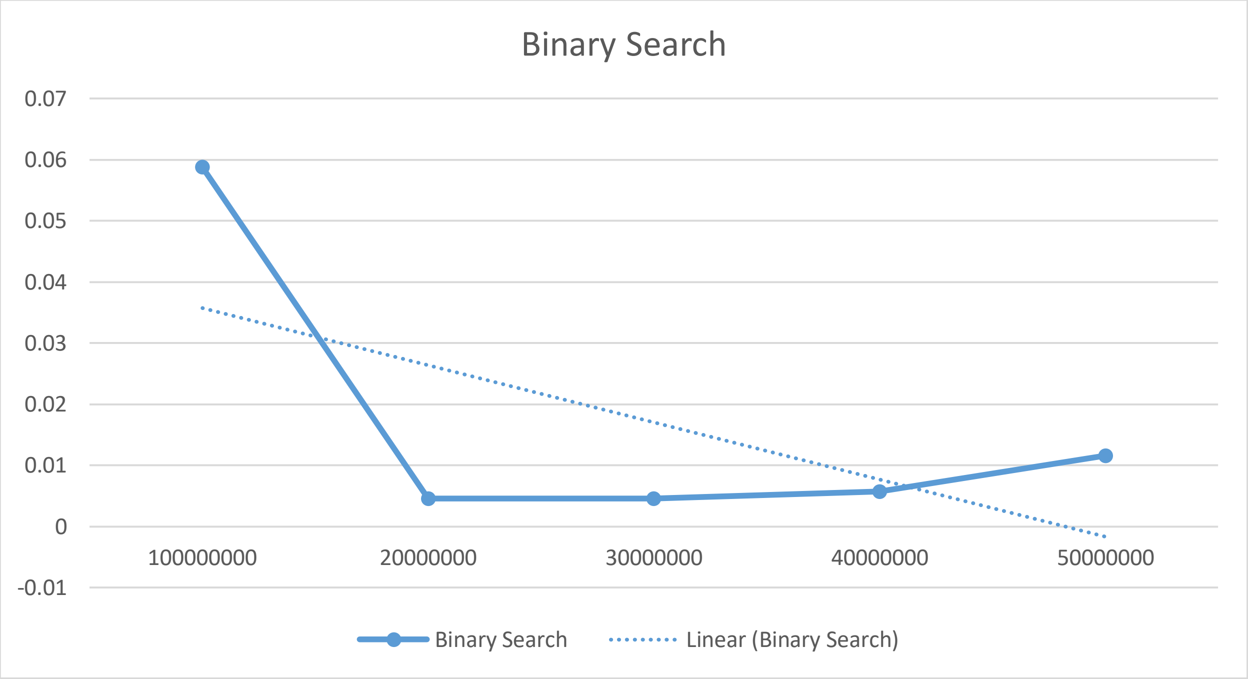


Combined:

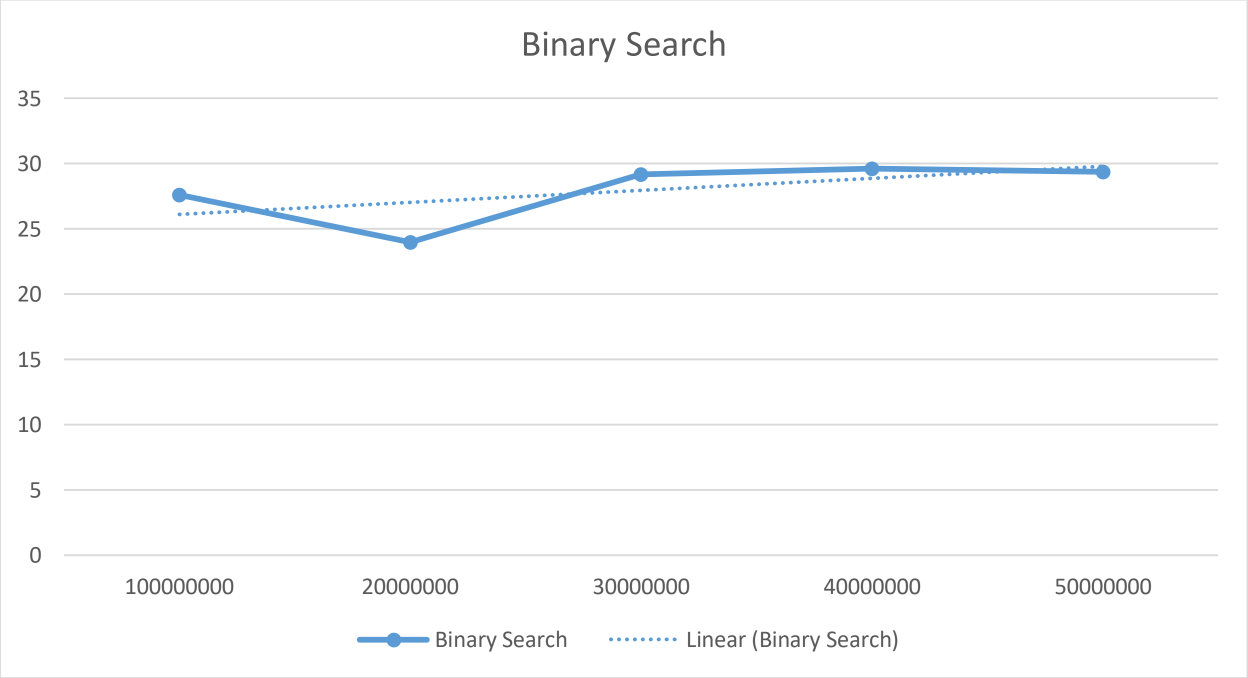


Binary search:

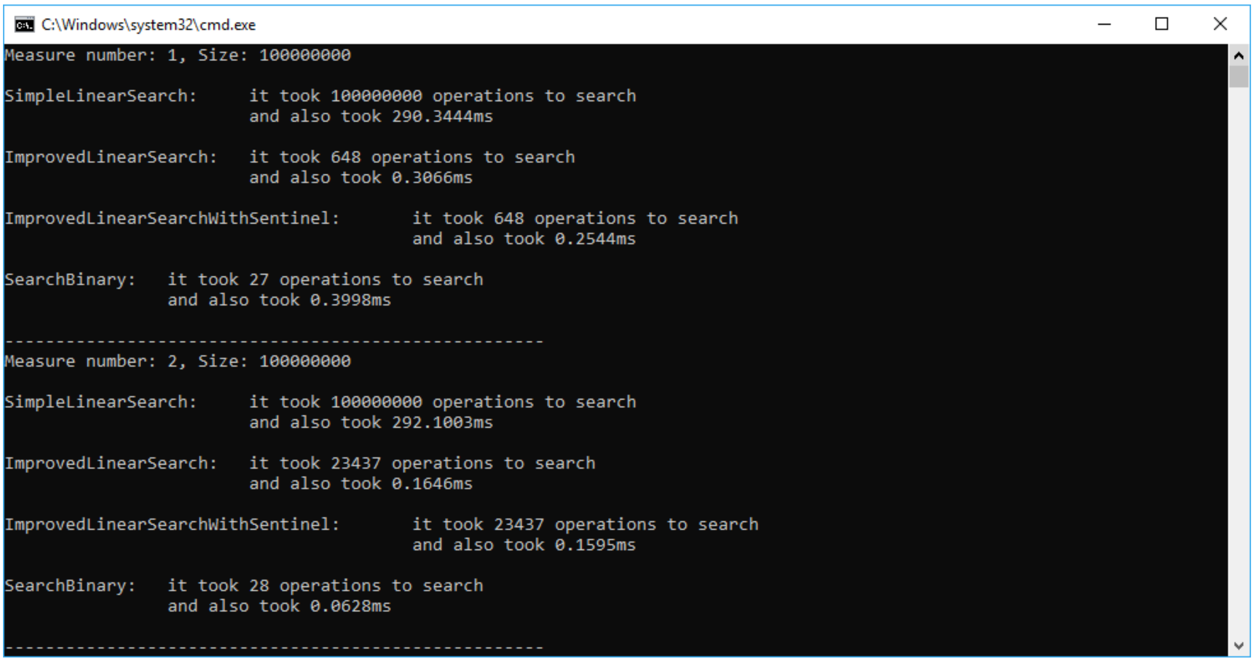
Time(ms) vs array size:



Operations vs array size:



Program screenshot:



```
Measure number: 1, Size: 100000000

SimpleLinearSearch:     it took 100000000 operations to search
                        and also took 290.3444ms

ImprovedLinearSearch:   it took 648 operations to search
                        and also took 0.3066ms

ImprovedLinearSearchWithSentinel:       it took 648 operations to search
                                        and also took 0.2544ms

SearchBinary:   it took 27 operations to search
                and also took 0.3998ms

-----------------------------------------------------
Measure number: 2, Size: 100000000

SimpleLinearSearch:     it took 100000000 operations to search
                        and also took 292.1003ms

ImprovedLinearSearch:   it took 23437 operations to search
                        and also took 0.1646ms

ImprovedLinearSearchWithSentinel:       it took 23437 operations to search
                                        and also took 0.1595ms

SearchBinary:   it took 28 operations to search
                and also took 0.0628ms

-----------------------------------------------------
```

Conclusion: We worked with 4 search algorithms and it is obvious that LinearSearch algorithm is the worst according to the data and graphs, complexity is always O(n) and it takes more time to search depending on the size of the array. Improved Linear search algorithms make the same amount of operations to find an element, but Improved linear search with sentinel works much faster. Binary Search wins here with complexity O(lg n) - the minimum number of operations, incredible speed comparing to linear search algorithms.