Title: Algorithm Efficiency and Sorting

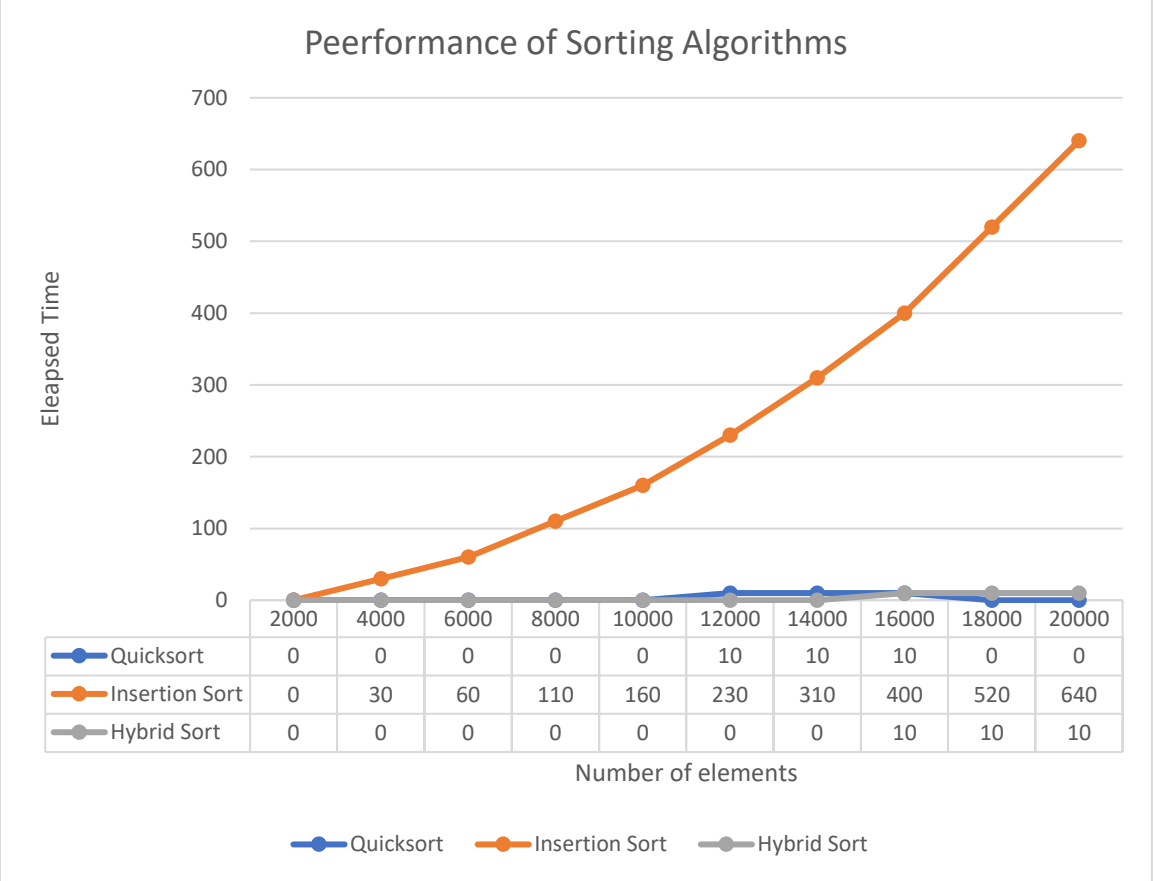
Author: Bora Fatih Kazancı

```
Array Size      Time Elapsed    compCount               moveCount
2000            0 ms            12361                   43080
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
2000            0 ms            989678          991677
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
2000            0 ms            10750                   34604
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
4000            0 ms            23816                   83448
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
4000            30 ms           4013609         4017608
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
4000            0 ms            20726                   66872
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
6000            0 ms            45159                   153477
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
6000            60 ms           9074747         9080746
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
6000            0 ms            40357                   128104
--------------------------------------------------------------------------------
```

```
Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
8000            0 ms            55034                   189099
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
8000            110 ms          15993004                16001003
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
8000            0 ms            48803                   155841
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
10000           0 ms            69614                   238839
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
10000           160 ms          25149970                25159969
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
10000           0 ms            62011                   197825
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
12000           10 ms           95996                   323985
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
12000           230 ms          36027366                36039365
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
12000           0 ms            86802                   274600
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
14000           10 ms           112903                  380709
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
14000           310 ms          48704856                48718855
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
14000           0 ms            102059                  322741
--------------------------------------------------------------------------------
```

```
Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
16000           10 ms           125517                  424551
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
16000           400 ms          63383545                63399544
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
16000           10 ms           113121                  358223
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
18000           0 ms            151199                  507594
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
18000           520 ms          81426904                81444903
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
18000           10 ms           137305                  433110
--------------------------------------------------------------------------------



Part a - Time analysis of Quick Sort
Array Size      Time Elapsed    compCount               moveCount
20000           0 ms            167711                  563133
--------------------------------------------------------------------------------
Part b - Time analysis of Insertion Sort
Array Size      Time Elapsed    compCount               moveCount
20000           640 ms          100631685               100651684
--------------------------------------------------------------------------------
Part c - Time analysis of Hybrid Sort
Array Size      Time Elapsed    compCount               moveCount
20000           10 ms           152671                  481734
--------------------------------------------------------------------------------
```

Graphical Analysis



For quicksort, the worst case is O (n^2), however, because of the number of elements and the speed of Dijkstra machine, algorithm has nearly a straight line. For the insertion sort, the worst case is O (n^2) like quick sort. However, insertion sort is slower than quick sort, which means we can consider quick sort instead of insertion sort while working with vast number of elements. The hybrid sort, which is mix of quick sort and insertion sort, is pretty good with small number of elements, because it uses insertion sort when array size is less than or equal to 10. However, we have worked with big numbers and it cause the use of quick sort at the end, and if we looked at size 16000, we could see that their working time is equal.

Hybrid sort algorithm has advantages with small number due to work logic I mentioned above. However, at big size of arrays, it's running time should be same as quick sort. Hence, hybrid sort algorithm could be pretty useful when working with both small and big sized arrays.