*Name: Ziad Sherif Muhammed* *BN:27* *Sec:1*

Design and Analysis of Algorithms

Assignment #2

## 1-Selection Sort

Selection sort is an**in-place comparison sorting algorithm**. It has an O (n2) time complexity, which makes it inefficient on large lists, and generally performs worse than the similar insertion sort.

## 2-Insetions Sort

Insertion sort is a simple sorting algorithm that works like the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part

## 3-Merge Sort

Merge sort is a divide-and-conquer algorithm based on the idea of breaking down a list into several sub-lists until each sub-list consists of a single element and merging those sub-lists in a manner that results into a sorted list.

## 4-Qucik Sort

Like [Merge Sort](https://www.geeksforgeeks.org/merge-sort/), Quicksort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quicksort that pick pivot in different ways (In this implementation using randomized pivot which gives better complexity).

## 5-Hybrid Sort

it’s a composite sort, which contains two different sorting (Quick and insertion sort). It’s used insertion sort when size of data is small while if too large, used quick sort till reach the partitioning`s size is small, so it used insertion sort. This sorting nearly O(NlogN).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1000 | 5000 | 10000 | 50000 | 75000 | 100000 | 500000 |
| Selection(ms) | 6.064 | 155.901 | 609.599 | 14890.213 | 34892.993 | 62683.854 | 1559613.793 |
| Insertion(ms) | 2.064 | 39.35 | 158.318 | 3643.579 | 8455.052 | 14497.049 | 373071.383 |
| Merge(ms) | 0 | 0 | 0 | 15.659 | 31.278 | 46.866 | 203.144 |
| Quick(ms) | 0 | 0 | 0 | 15.653 | 15.67 | 31.37 | 141.689 |
| Hybrid(ms) | 1.014 | 1.994 | 3.996 | 20.455 | 28.923 | 39.893 | 208.101 |

Random data

Sorted data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1000 | 5000 | 10000 | 50000 | 75000 | 100000 | 500000 |
| Selection(ms) | 2.991 | 66.821 | 271.346 | 7063.268 | 16181.471 | 26763.241 | 723321.788 |
| Insertion(ms) | 0 | 0 | 0 | 0 | 0.999 | 0 | 4.020 |
| Merge(ms) | 1.037 | 1.995 | 3.008 | 15.431 | 21.941 | 28.929 | 143.188 |
| Quick(ms) | 0 | 0.998 | 0.998 | 7.942 | 13.963 | 19.947 | 144.655 |
| Hybrid(ms) | 0.998 | 1.995 | 3.988 | 18.950 | 27.971 | 40.939 | 222.126 |