

Assignment 2

Algorithms and Data Structures 1 (1DL210) 2021

Hooman Asadian

Sarbojit Das

September 23, 2021

31st group:

Samet Burak Türk

I got no invitation and I couldn't find anyone without a group for now the group has 1 person.

Comparison:

Insertion sort: This is the most naive approach among these 3 sorting algorithm and have an average $\Theta(n^2)$ time complexity which is when compared to other algorithms time complexity is really bad and inefficient but for sorted algorithms will perform $\Theta(n)$ complexity which not bad and can be used to check if a array is sorted. Another advantage is sorted part is always on the left side of the array that means if a new item is added to end of the array it won't intervein with sorted part and can be used for an array that changes while sorting is taking place.

Quick sort: with average of $\Theta(n \log(n))$ time complexity quick sort is a reliable sorting algorithm. Even though it has similar average time complexity with heap sort on python a comparison made with `time.perf_counter()` proved that quick sort is faster for arrays that are smaller when run on python.

Heap sort: similar to quick sort average time complexity is $O(n \log(n))$ but in addition worst case is also the same so if there is a case where you don't want to risk of having to deal with worst case heap sort is better than quick sort also on python heap sort used less recursion compared to quick sort if there is `recursionlimit` for another function and you don't want to change it with `sys.setrecursionlimit()` then heap sort is better because it will go less deeper than quick sort.

Instructions:

Import `sort.py`, `rangem.py` and `test.sh`

When bashing `test.sh` pass an integer as the argument. For example:

```
bash test.sh 76
```