

## Homework 4

Problem 1

- a) My bubble sort can be found in "bubble.py".
- b) Worst case and average case would have a time complexity of  $O(n^2)$  because they are going through two full loops. And the best case would be if the array is already sorted so it would go once for every element ~~and~~ in the ~~first~~<sup>inner</sup> loop and it would enter the ~~other~~ loop only one time and because of that it would have a time complexity of  $O(n)$ .
- c) Stable sorting algorithms are: Insertion, Merge and Bubble sort. Heap sort is not a stable sorting algorithm. Stable sorting algorithms maintain stability by swapping only if one element is less than another ( $arr[i] < arr[j]$ ), and if they are equal ( $arr[i] = arr[j]$ ) they keep their position.
- d) From the mentioned only Insertion and Bubble sort are adaptive, because if they are given an already sorted array they wouldn't need any swapping and they would just go through the array. But that is not the case for Heap and Merge sort, they both won't get any faster even if the array is already sorted.

## Problem 2

- a) The normal heap sort can be found in "normalHeapSort.py"
- b) The bottom up heap sort can be found in "bottomUp.py"
- c) From the "plot.xlsx" we can see that the bottom up heap sort is faster than the normal heap sort.