**Homework 4**

**By Farzan Ali Khan**

**Problem 4.1**

1. Do

swapped = false

for i = 2 to length(A)-1

inclusive do:

if A[i] > A[i+1] then

/ ∗ swap them and remember something changed ∗/

swap( A[i-1], A[i] )

swapped = true

end if

end for

until not swapped

1. The Time complexity for worst case :

The number of comparisons made is (n-1 +n-2 + . . .+1). We go into each if statement and swap them.

Since the sum of the first n elements is = n(n+1)/2, the sum of the first n-1 elements is = n(n-1)/2

Hence the complexity becomes O(

The time complexity for the best case is O(n) since no swaps are made.

The time complexity for the average case remains the same as the worst case i.e O(.

1. Insertion sort, Merge sort and bubble sort are stable sorting algorithms since they use a comparison like:

if A[i] > A[i+1] then

.

.

End

If the keys are the same the one that comes before stays in that position.

For heap sort this is not the case. We lose the order in which the original elements are when using the heapify method. The order by which they are printed may depend on which node they are assigned to when using the heapify method and may change hence heap sort is not a stable sorting algorithm.

1. Insertion sort, Bubble sort and Heap sort are adaptive since before any operation is performed on the elements, a comparison is made if there is a need to actually do anything. This is not the case for Merge sort since before the comparisons are made, the elements are already split into the simplest sub-problems.