CS2223
HW#4

DUE: Monday, November 20

1. (5 points) Call an array of integers *partially sorted* if all the negative number precede all the positive numbers. That is, the list $(-34,-17,-17,-55,-2,-85,955,16,10978)$ is partially sorted. Describe an algorithm to partially sort an array. Your algorithm should execute in worst case in $\Theta(n)$. Pseudo-code suffices; you don't need to code your algorithm.

2. (14 points) Another local transposition algorithm to sort $A[1..n]$ is BUBBLESORT.

```
BUBBLESORT(A)
    for i ← 1 to n do                          ▷ A[1] ≤ ... ≤ A[i]
        for j ← n downto i+1 do
            if A[j−1] > A[j] then swap(A[j-1],A[j])
```

a Letting the benchmark operation be the pairwise comparisons $A[j-1] > A[j]$, what is the best case execution time of BUBBLESORT? You may use asymptotic notation, but show your work.

b Letting the benchmark operation be the pairwise comparisons $A[j-1] > A[j]$, what is the worst case execution time of BUBBLESORT? You may use asymptotic notation, but show your work.

c Letting the benchmark operation be the pairwise comparisons $A[j-1] > A[j]$, what is the average case execution time of BUBBLESORT? You may use asymptotic notation, but show your work.

d The invariant $A[1] \le ... \le A[i]$ can be used to **help** prove that the program correctly sorts the numbers. Namely, when the program is finished, $A[1] \le ... \le A[n]$. But what else has to be proved to establish that BUBBLESORT works correctly? Namely, the invariant $A[1] \le ... \le A[i]$ is necessary but not sufficient. What else do we have to establish?

3. (6 points) Given a weighted graph $G = (V, E)$ with $w: E \to \mathbb{R}^+$, we want to construct an acyclic subgraph $H = (V, E')$, $E' \subseteq E$ of maximal weight, where the weight of a graph is the sum of the weights of its edges, $w(H) = \sum_{e \in E'} w(e)$. Describe a polynomial time algorithm (pseudo-code suffices) to solve this problem.

CS2223
HW#4 SOLUTIONS

1. $temp \leftarrow A[1]$

   $A[1] \leftarrow 0$

   $A[\text{Partition}(A, 1, n)] \leftarrow temp$

2. ***a***, ***b***, ***c*** $\sum_{1 \le i \le n} \sum_{i+1 \le j \le n} 1 = \sum_{1 \le i \le n} (n - i) = \sum_{1 \le i \le n} n - \sum_{1 \le i \le n} i = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$

d We also need to establish that the final *A* is a permutation of the initial *A*. That is, that the final values are exactly the same as the initial values, though perhaps in a different order.

3. BUILD-MAX-HEAP(*E*)

   $T \leftarrow \varnothing$

   while $E \neq \varnothing$

      $e \leftarrow$ EXTRACT-MAX(*E*)

      if $T \cup \{e\}$ acyclic **then** $T \leftarrow T \cup \{e\}$

   return *T*