CS 4349.003 Homework 2

Due Wednesday September 11th on eLearning

September 4, 2019

Please answer the following **2** questions.

1. Using Θ-notation, provide asymptotically tight bounds in terms of n for the solution to each of the following recurrences. Assume each recurrence has a non-trivial base case of $T(1) = \Theta(1)$. For example, if asked to solve $T(n) = 2T(n/2) + n$, then your answer should be $\Theta(n \log n)$. **Give a brief explanation for each solution.**

   (a) $T(n) = 5T(n/2) + n$

   (b) $T(n) = 4T(n/2) + n^2$

   (c) $T(n) = T(n/4) + T(n/2) + n$

   Consider the procedure $\textsc{StoogeSort}(A[1..n])$ which sorts the given array $A[1..n]$.

```
StoogeSort(A[1..n]):
  if n = 2 and A[1] > A[2]
      swap A[1] ↔ A[2]
  else if n > 2
      m ← ⌈2n/3⌉
      StoogeSort(A[1..m])
      StoogeSort(A[n − m + 1..n])
      StoogeSort(A[1..m])
```

   (d) State a recurrence for the running time of $\textsc{StoogeSort}(A[1..n])$. Your solution should have the form $T(n) = rT(n/c) + f(n)$ where r and c are constants and $f(n)$ is a (very) simple function. You should assume lines like $\textsc{StoogeSort}(A[i..j])$ only incur constant time overhead in addition to the time spent inside the recursive call. Don't forget to justify your answer!

   (e) Solve your recurrence from part (d) in order to determine the running time of $\textsc{StoogeSort}(A[1..n])$. As before, you should briefly explain your solution.

   (f) **(Extra credit worth half a question.)** Prove $\textsc{StoogeSort}(A[1..n])$ actually sorts its input.

2. An ***inversion*** in an array $A[1..n]$ is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$. The number of inversions in an n-element array is between 0 (if the array is sorted) and $\binom{n}{2}$ (if the array is sorted backward).

   (a) Consider a procedure CountAndMerge($A[1..n], m$) that takes as input an array $A[1..n]$ and an integer m such that $0 \le m \le n$, $A[1..m]$ is sorted, and $A[m+1..n]$ is sorted. The procedure should 1) count the number of inversions (i, j) such that $1 \le i \le m$ and $m + 1 \le j \le n$; 2) shuffle the elements of $A[1..n]$ so they are in sorted order; and 3) return the number of inversions counted.

   Describe and analyze an implementation of CountAndMerge($A[1..n], m$) based on the Merge($A[1..n], m$) procedure in Erickson Figure 1.6. You should justify correctness of your implementation, preferably using a proof by induction.

   (b) Consider a procedure CountAndSort($A[1..n]$) that takes as input an array $A[1..n]$ whose elements can be in any order. The procedure should 1) count the number of inversions in $A[1..n]$; 2) shuffle the elements of $A[1..n]$ so they are in sorted order; and 3) return the number of inversions counted.

   Describe and analyze an implementation of CountAndMergeSort($A[1..n]$) based on the MergeSort($A[1..n]$) procedure in Erickson Figure 1.6. Again, justify correctness of your implementation. You may assume your answer to part (a) is correct.