COMP285: Analysis of Algorithms
North Carolina A&T State University — Homework #3
Dr. Allison Sullivan — Due: September 17, 2019

Homework #3

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Finding A Match

Your working at a match making company and your job is to develop a more efficient algorithm for returning matches. You receive a specific person's match score as an integer, x, as well as a list of all other's match scores as an array of integers A. This array is maintained in sorted order. A match is someone who has the same score or if a perfect match does not exists, you want to return the largest score smaller than x. Write an efficient algorithm to return the matched value for x. Write the recurrence relation for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method or a recursion tree).

Problem 2: Can it be done?

For each of the following tasks, either **explain briefly how you would accomplish it**, or else **explain why it cannot be done.**

(a) Find the maximum of an unsorted array of length n in time $O(1)$.

(b) Find the maximum of an unsorted array of length n in time $O(n \log(n))$.

(c) Given an array A of length n which contains only integers between 1 and n, sort A in time $O(n)$.

(d) Given an array A of length n which contains arbitrary comparable elements, sort A in time $O(n)$.

(e) Given an array A of length n, find the $(n/3)^{rd}$ smallest element in time $O(n)$.

Problem 3: LexicoSort

A lexicographic sorting algorithm aims to sort a given set of n strings into lexicographically ascending order (in case of ties due to identical strings, then in non-descending order). For each of the following, describe your algorithm clearly, and analyze its running time.

(a) Give an O(n) lexicographic sorting algorithm for n strings where each string consists of exactly one letter from the set $a - z$ (that is, each string is of length 1). (Hint: think about the CountingSort)

(b) Give an $O(mn)$ lexicographic sorting algorithm for n strings where each string contains letters from the set $a - z$ and is of length at most m. (Hint: you'll need to pad words of $length < m$.)