Homework 3

OPTIONAL PROBLEMS
(No due date)

1 Written Problems

1.1 Longest Increasing Subset

NOTE: this is a challenge problem
Given an array of integers, find the length of the longest increasing subsequence of these integers. (This can be done with dynamic programming in O(nlogn) time!!) For example,

1. $[0, 3, 6, 2, 10, 1, 5, 33]$ would return 5, because the longest increasing subsequence, $[0, 3, 6, 10, 33]$ is of length 5

2. $[14, 2, 15, 11]$ would return 2, because the longest increasing subsequences, $[14, 15]$, $[2, 11]$ and $[2, 15]$ are all of length 2

1.2 Serena Williams' Fan Club Meeting

You know that at the club meeting, there is only one real Serena Williams. Within a group of n girls, *Serena Williams* is defined as a girl who is known by everyone, but knows no one. At the club meeting, you are only allowed to ask questions of the form "Excuse me, but do you know *that* girl over there?" Your job is to determine whether Serena Williams is actually in the group. Clearly you can do this by asking each of the n girls about each of the other $n-1$ girls, a total of $n(n-1)$ questions. But you'd like to do much better!! Note: If a group has size $n = 1$, you should assume the sole member of the group is Serena Williams (after all, she knows no one, and everyone knows her). Note 2: As creepy as it may sound in real-life, for this question you should not assume that girl A knowing girl B implies girl B knows girl A, even if both are not Serena Williams!

(a) Explain why there can be at most one *Serena Williams* in this club meeting, and describe a group of size n (for every $n > 1$) in which there's no such person.

(b) Suppose you ask A whether they know B, and A says "Yes." What (if anything) can you conclude about the identities of A and B? What if A says "No"? What can you conclude then (if anything)?

(c) Describe an algorithm that, given a group of n girls, either finds Serena Williams or determines that Serena Williams isn't in the group of fans using $O(n)$ questions. A short paragraph explanation will suffice.

(d) Prove by induction that your algorithm uses $O(n)$ questions. By this we mean use the same logical structure as an inductive proof in your written explanation (i.e. with a base case, inductive step, etc.). In order to use $O(n)$

questions, you may ask only a constant number of questions for each girl in the fan club meeting. You will lose points if you don't use induction.

2 Python Problems

2.1 Climbing Stairs

Given a set of stairs comprised of n steps, and that you can climb 1, 2, or 3 steps at a time, write a function that returns the number of ways to climb that set of stairs. For example,

1. climber(2) would return 2, because the sequence of steps could be $[1, 1]$ or $[2]$
2. climber(3) would return 4, because the sequence of steps could be $[1, 2]$, $[2, 1]$, $[3]$, or $[1, 1, 1]$

You do not need to return the possible sequences of steps, just the number.

2.2 Climbing Stairs++

Modify your climber(n) function to print out the possible sequences of steps, such that climber(3) would print out "$[1, 2]$, $[2, 1]$, $[3]$, $[1, 1, 1]$"