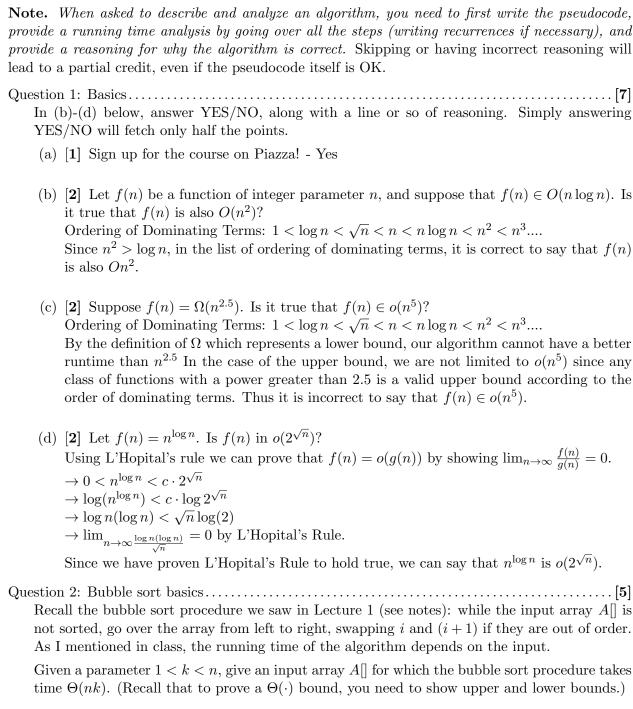
CS 6150: HW1 – Data structures, recurrences

Pranav Rajan, Collaborators: Adithya Badideymurali, Ryan Howell, Frost Office Hours
Submission date: Monday, Sep 13, 2021 (11:59 PM)

This assignment has 6 questions, for a total of 50 points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question	Points	Score
Basics	7	
Bubble sort basics	5	
Deletion in prefix trees	6	
Binary search and test pooling	10	
Recurrences, recurrences	16	
Dynamic arrays: is doubling important?	6	
Total:	50	



Show how to modify the data structure so that this can be avoided. More formally, if S is the set of words remaining after a sequence of add/delete operations, we would like to ensure space utilization that is the same (possibly up to a constant factor) of the space needed to store only the elements of S. If your modifications impact the running time of the add, query, and delete operations, explain how.

It turns out that if only a "few" people have the disease, this is much better than testing all n people.

- (a) [4] Suppose we know that exactly one of the n people has the disease and our aim is to find out which one. Describe an algorithm that runs in time $O(\log n)$ for this problem. (For this part, pseudocode suffices, you don't need to analyze the runtime / correctness.)
- (b) [6] Now suppose we know that exactly two of the n people have the disease and our aim is to identify the two infected people. Describe and analyze (both runtime and correctness) an algorithm that runs in time $O(\log n)$ for this problem.

- (a) [4] $T(n) = 3T(n/3) + n^2$. As the base case, suppose T(n) = 1 for n < 3.
- (b) [6] T(n) = 2T(n/2) + T(n/3) + n. As the base case, suppose T(0) = T(1) = 1.
- (c) [6] $T(n) = 2(T(\sqrt{n}))^2$. As the base case, suppose T(1) = 4.

Suppose we consider an alternate implementation, where the array size is always a multiple of some integer, say 32. Every time the add procedure is called and the array is full (and of size n), suppose we create a new array of size n + 32, copy all the elements and then add the new element.

For this new add procedure, analyze the asymptotic running time for N consecutive add operations.