

Homework 2

Jason Jiang
CS 577

February 7, 2019

Question 1. The students of the University of Disclitown usually line up in front of a small canteen to buy their lunch. They make a perfect line so a student can see what is available in the canteen only if everybody in front of them is shorter than themselves. For simplicity, we assume that the heights of the students are positive integers. We model the line of students as an array of heights. For example **[3, 4, 2, 4, 8]** means that the student closest to canteen has height 3 the next student has height 4 and the last student in the line has height 8. The three bold students in the array can see the canteen. Consider the specification and procedure in Algorithm 3.

- (a) Fill in the specification of procedure foo of Algorithm 3. The blanks are indicated by???. Note that the output depends on t.
(b) Prove that Algorithm 3 is correct.

- a)

Input: A list of n positive integers with $n \geq 0$, an integer $t \geq 0$

Output: The number of students in a part of an line who can see the canteen given the maximum height of previous students is t.

- b)

Proof for procedure FOO:

Base Case: when $n = 0$ the program foo will return 0, which match the value we expected that when there is nobody in line there will be nobody can see the canteen.

We assume the program is correct until $n=k$, where $k \geq 0$.

For $n=k+1$, we will have:

Because the program is correct until k, so when $n=k+1$, there is an additional call of FOO of $FOO(A[k+1], t)$ **(1)**;

if $A[k+1] > t$ the program will return, as line 9, $1 + FOO(A[], A[k+1])$ **(2)**. Because in (2) the A becomes empty the recursion terminates and return 0 as the base case. Hence the (1) will return 1, and will be added to the result when $n=k$. It draws the correct result.

if $A[k + 1] < t$ the program will run line 11, simply return $FOO(A[], t)$ and because A is empty it returns 0. Then the whole recursion terminates and return the same value as $n=k$. It is correct as we expected.

Hence, the algorithm is correct for $n=k+1$, and produces correct answer as the program terminates.

The program terminates. Because when $n!=0$, the program recursively calls FOO with length $n-1$ and the n will eventually becomes 0 and terminates the whole program.

Hence the program FOO is correct.

Because procedure just return the value of FOO with a starting maximum of 0, it is correct as long as FOO is correct.

Hence, the algorithm 3 is correct.