

CSE 311: Foundations of Computing I

Homework 6 (due Wed, Nov 16, at 6:00 PM)

Directions: Read the Grading Guidelines page before you start working on your solutions. Write up carefully argued solutions to the following problems. The first task is to be complete and correct. The more subtle task is to keep it simple and succinct. Your solution should be clear enough that it should explain to someone who does not already understand the answer why it works. You may use any results proven in lecture without proof. Anything else must be argued rigorously. Unless otherwise specified, all answers are expected to be given in closed form.

1. Keeping Track of the Leftovers (15 points)

Prove the following inequality for all integers $n \geq 2$:

$$\sum_{i=2}^n \frac{1}{(i-1)i} < 1$$

Hint: What is the sum exactly?

2. Runtime... Better Go Catch It! (20 points)

Let $c > 0$ be an integer. The following recursive definition describes the running time of a recursive algorithm.

$$\begin{aligned} T(0) &= 0 \\ T(n) &\leq c && \text{for all } n \leq 20 \\ T(n) &= T\left(\left\lfloor \frac{3n}{4} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{5} \right\rfloor\right) + cn && \text{for all } n > 20 \end{aligned}$$

Prove by strong induction that $T(n) \leq 20cn$ for all integers $n \geq 0$.

Hint: The only fact about $\lfloor \cdot \rfloor$ that you will need is that when $x \geq 0$, $\lfloor x \rfloor$ is an integer, and $0 \leq \lfloor x \rfloor \leq x$.

3. Happily Ever After (15 points)

Let S be the set defined as follows.

Basis Step: $5 \in S$; $9 \in S$;

Recursive Step: if $x, y \in S$, then $x + y \in S$.

Show that there is an integer a such that for every integer $n \geq a$, it holds that $n \in S$.

Hint: Strong induction is the right tool here since the quantifier is not over S .

4. Putting car into Reverse is rac (15 points)

For strings over alphabet Σ , we defined the reversal operation by $\varepsilon^R = \varepsilon$ and for any $w \in \Sigma^*$ and $a \in \Sigma$

we have $(wa)^R = aw^R$. Prove that $(x \bullet y)^R = y^R x^R$. for all $x, y \in \Sigma^*$.
 You may use without proof the fact the $(x \bullet (y \bullet z)) = (x \bullet y) \bullet z$ for all $x, y, z \in \Sigma^*$.

5. Binary Strings (15 points)

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

- (a) [5 Points] Binary strings where every occurrence of a 1 is immediately followed by a 0.
- (b) [5 Points] Binary strings with an even number of 1s.
- (c) [5 Points] Binary strings with equal number of 0s and 1s.

6. Proving BST Insertion Works! (20 points)

Consider the following definition of a (binary)Tree:

Bases Step: Nil is a Tree.

Recursive Step: If L is a Tree and R is a Tree and x is an integer, then $\text{Tree}(x, L, R)$ is a Tree.

The standard *BST insertion* function can be written as the following:

$$\begin{aligned} \text{insert}(v, \text{Nil}) &= \text{Tree}(v, \text{Nil}, \text{Nil}) \\ \text{insert}(v, \text{Tree}(x, L, R)) &= \begin{cases} \text{Tree}(x, \text{insert}(v, L), R) & \text{if } v < x \\ \text{Tree}(x, L, \text{insert}(v, R)) & \text{otherwise.} \end{cases} \end{aligned}$$

Next, define a program **less** which checks if an entire BST is less than a provided integer v :

$$\begin{aligned} \text{less}(v, \text{Nil}) &= \text{true} \\ \text{less}(v, \text{Tree}(x, L, R)) &= x < v \text{ and } \text{less}(v, L) \text{ and } \text{less}(v, R) \end{aligned}$$

Prove that for all $b \in \mathbb{Z}$, $x \in \mathbb{Z}$ and all trees T , if $\text{less}(b, T)$, and $b > x$, then $\text{less}(b, \text{insert}(x, T))$.

In English, this means that, given an upper bound on the elements in a BST, if you insert something that meets that upper bound, it is still an upper bound.

You should use structural induction on T for this question, but there are a few tricky bits that are worth pointing out up-front:

- You are proving an implication *by induction*. This means, in your Base Case, you assume the first part and prove the second one.
- Because of this, there will be two implications going on in your Induction Step. This can be very tricky. You will assume *both* your IH and the left side of what you're trying to prove. You will end up needing to use both of them at some point in your proof.
- This is the most difficult proof we have given you to date. It would be a mistake to start it on the last day.

7. Extra Credit: Magical Pebbles (-NoValue- points)

Consider an infinite sequence of positions $1, 2, 3, \dots$ and suppose we have a pebble at position 1 and another pebble at position 2. In each step, we choose one of the pebbles and move it according to the following rule: Say we decide to move the pebble at position i ; if the other pebble is not at any of the positions $i + 1, i + 2, \dots, 2i$, then it goes to $2i$, otherwise it goes to $2i + 1$.

For example, in the first step, if we move the pebble at position 1, it will go to 3 and if we move the pebble at position 2 it will go to 4. Note that, no matter how we move the pebbles, they will never be at the same position.

Use induction to prove that, for any given positive integer n , it is possible to move one of the pebbles to position n . For example, if $n = 7$ first we move the pebble at position 1 to 3. Then, we move the pebble at position 2 to 5. Finally, we move the pebble at position 3 to 7.