Homework 6: Induction

Due date: Wednesday Feburary 22nd at 10 PM
If you work with others (and you should!), remember to follow the collaboration policy outlined in the syllabus.

In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.

We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

In order to assist with the transition from formal proofs to English proofs, we've published a style guide on the website containing some tips. This guide contains references to proof materials that we haven't taught yet, so don't worry if some of these terms are unfamiliar.

Finally, be sure to read the grading guidelines for more information on what we're looking for.

To help you pace yourself on this assignment, we've divided the assignment into two parts and estimate when you should finish the first part. For submission purposes, you may ignore these recommendations. The whole assignment (both parts) has a single gradescope box with a duedate of Wednesday May 18th.

1. Put One Foot In Front of the Other [9 points]

Do the concept check for this week on gradescope.

2. Proof by contradiction [10 points]

Use proof by contradiction to show that $\sqrt{19}$ is irrational.

You may use the following fact without needing to prove it: For every prime number p and integers a, b: if $p|ab$ then $p|a$ or $p|b$.

You **MUST** use induction for the proofs for the rest of this assignment (except the feedback). You may use any appropriate version of induction (e.g. weak or strong or structural). Remember to define a predicate $P()$ as part of your proof.

3. Running Times [20 points]

You wrote a piece of recursive code. On an input of size n (for $n \in \mathbb{Z}^+$), your function takes $T(n)$ time to run, where:

$$T(n) = 5n \qquad \text{if } 1 \leq n \leq 4$$
$$T(n) = T(\lfloor n/2 \rfloor) + T(\lfloor n/4 \rfloor) + 5n \qquad \text{for all } n > 4$$

In the definition above, $\lfloor x \rfloor$ is the "floor" function, it returns the greatest integer at most x.
For example: $\lfloor 3.2 \rfloor = 3$, $\lfloor 3.7 \rfloor = 3$, $\lfloor 3 \rfloor = 3$.

Show that for all $n \in \mathbb{N}$ with $n \geq 1$, $T(n) \leq 20n$

Caution: Your $T()$ is only defined to take positive integers as input. $T(1.5)$ is not defined!

Hint 1: Notice that while $T()$ is defined with equality, you are only proving an inequality.
Hint 2: The only fact about the floor function you will need is $\lfloor x \rfloor$ is an integer and $1 \leq \lfloor x \rfloor \leq x$.

4. These are pretty long str1111ngs [20 points]

Let 1^n mean a string of n ones. Let S be the set of strings defined as follows:

Basis Steps: $1^3 \in S$, $1^{11} \in S$
Recursive Step: If $1^x, 1^y \in S$ then $1^x \cdot 1^y \in S$ where $\cdot$ is string concatenation.

Show that, for every integer $n \geq 20$ the set S contains the string 1^n.

Caution: Structural Induction is not the best tool for this problem. Structural induction shows $\forall x \in S(P(x))$. You're analyzing what the elements of S are in this problem, not proving a predicate holds for all elements of S.

5. The Apple Doesn't Fall Far From The... `Tree` [20 points]

In CSE 143, you saw a recursive definition of trees. That definition looks a little different from what we saw in class.

The following definition is analogous to what you saw in 143. We'll call them `JTrees`.

Basis Step: `nil` is a `JTree`.
Recursive Step: If L, R are `JTrees` then (`data`, L, R) is also a `JTree`.

Show that for all `JTree`: if they have $c - 1$ copies of `data` then they have c copies of `nil`.

Remark: You're effectively showing here that a binary tree with $c - 1$ nodes has c `nil` child pointers.

6. Feedback [1 point]

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment (excluding any extra credit questions, if applicable)? Report your estimate to the nearest hour.
- Which problem did you spend the most time on?
- Any other feedback for us?