Homework 6: More Induction

Due date: Wednesday November 12th at 11:59 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 writing 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.

1. (Set)Theoretically speaking... [10 points]

Define the sets $A := \{(x, y) \in \mathbb{R}^2 \mid y \geq e^x\}$, $B := \{(x, y) \in \mathbb{R}^2 \mid x \leq 0\}$, $C := \{(x, y) \in \mathbb{R}^2 \mid y \geq |1 + x|\}$, where each set is a region on the xy-plane. Prove that $A \subseteq B \cup C$.
Hint: You may use the following facts:

- For every real number $x \geq 0$, $1 + x \leq e^x$.
- For any real numbers a, b such that $b \geq 0$, if $a \geq b$, then $a \geq |b|$.

2. An Empty Set [10 points]

Let A and B be sets. Write a proof by contradiction that shows: $\forall A \forall B[(A \setminus B) \cap (B \setminus A) = \emptyset]$. Be sure to explicitly mention any quantifiers in what you suppose at the start of your proof.

You **MUST** use induction for problems 3-6 unless the directions for a part say otherwise.
You may use any appropriate version of induction (e.g. weak or strong). Remember to define a predicate $P()$.

3. Boba, and repeat! [20 points]

Joey is hosting a Clairo-listening, home-made pasta eating, astrology reading, labubu accessorizing party and is preparing for it. He asks Rachel to make him some boba and matcha drinks. Rachel makes drinks in batches of exactly 3 cups of boba or exactly 8 cups of matcha. A single batch cannot contain both boba and matcha (obviously). Joey's friends are tooooo busy collecting labubus and contemplating injustices and only 14 have RSVP-ed. To ensure that Joey can be fiscally responsible but still provide the necessary number of drinks, prove that for every integer $n \geq 14$, there is some distribution of batches of boba and batches of matcha totaling n cups such that Rachel can make exactly n drinks.

4. f(unky) Induction [20 points]

Suppose we have the following recursively defined function,

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 4 & \text{if } n = 1 \\ (16n^2 - 16n) \cdot f(n-2) & \text{otherwise} \end{cases}$$

Use induction to prove that for all integers n with $n \geq 0$, $f(n) = 4^n n!$.
Recall that for a positive integer n, $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 1$, and that $0! = 1$.

5. What an Odd Tree... [20 points]

In this problem, we'll refer to the following definition of an `oddTree`:

Basis Step: If x is an **odd** integer $(\texttt{null}, x, \texttt{null})$ is an `oddTree`.
Recursive Step: If L, R are `oddTrees` and x is an **odd** integer, then (L, x, R) is also an `oddTree`.

We will also use the following recursively defined function that sums the values in an `oddTree`:

$$\texttt{sum}((\texttt{null}, x, \texttt{null})) = x$$
$$\texttt{sum}((L, x, R)) = x + \texttt{sum}(L) + \texttt{sum}(R) \qquad \text{for two arbitrary oddTrees } L, R$$

Show that the sum of the values in an `oddTree` is always odd. In other words, show that $\texttt{sum}(T)$ is odd for every `oddTree` T.

For this problem: You should use the definitions of odd/even directly in your answer. If you wish to use any theorems about the result of adding odd/even numbers, you must prove them in your work.

6. Real World: Some Simple Code [23 points]

Computer scientists write code, but if you want other people to use your code, you'll have to be able to explain why it works. In future classes (especially CSE 421), you'll do that with a proof.[1]

Let's take an example of code that returns the product of the first n even integers for inputs of $n > 0$.[2]

```
public static int multiplyEvens(int n) {
    if (n <= 0) {
        throw new IllegalArgumentException();
    }
    if (n == 1) {
        return 2;
    }
    return (2 * n) * multiplyEvens(n-1);
}
```

Call	Output	Reason
`multiplyEvens(1);`	2	2 = 2
`multiplyEvens(2);`	8	2 * 4 = 8
`multiplyEvens(3);`	48	2 * 4 * 6 = 48
`multiplyEvens(4);`	384	2 * 4 * 6 * 8 = 384

In this problem, you will write a correctness proof that will show that this code always returns the product of the first n even integers for inputs of $n > 0$. In other words that

$$\texttt{multiplyEvens(n)} = \prod_{i=1}^{n} 2i$$

Capital pi ($\prod$) is the product symbol which represents repeated multiplication, similar to $\sum$ which is for repeated summation. For example, $\prod_{k=1}^{3} 4k = 4 \cdot 8 \cdot 12$.

(a) Prove that the code produces the desired output. You must use induction for this problem. Be sure to start by defining your predicate $P()$. [20 points]

[1] In the real world, you won't do a full proof (unless you become a researcher), but you will still have to clearly explain what's going on in your code, and a proof is a good way to practice a careful explanation

[2] We took this coding problem from `Practice-It`; It was authored by Whitaker Brand (on 2019/09/19)

(b) Take a moment to reflect on the structure of the proof and the code. The code has a base case and a recursive case, which relies on the result for input $n-1$ to calculate the answer for input n. The proof will have a base case, and an inductive case from k to $k+1$. You do not have to write anything for this part [0 points]

(c) Imagine you were tasked with convincing someone the code snippet works. What would you do (write this proof? Run some test cases? A combination of those? Something else?)? How would your strategy change if this code snippet had a different structure (e.g., no recursion)? (3-4 sentences, but more is okay) [3 points]

7. Extra Credit: Induction Rocks!

You will submit this question to the separate gradescope box for "homework 6 extra credit."

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

For example, in the first step, if we move the stone at position 1, it will go to 3 and if we move the stone at position 2 it will go to 4. Note: no matter how we move the stones, 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 stones to position n. For example, if $n = 7$ first we move the stone at position 1 to 3. Then, we move the stone at position 2 to 5 Finally, we move the stone at position 3 to 7.