0 Overview

Instructions

Due Date: Sunday Mar 9 @ 11:59PM

Total points: 50

Your written solution may be either handwritten and scanned, or typeset. Either way, you must produce a PDF that is legible and displays reasonably on a typical PDF reader. This PDF should be submitted via autolab as WA3. You may submit as many times as you like, but only your last submission will be graded and should include the entire submission. You should view your submission after you upload it to make sure that it is not corrupted or malformed. Submissions that are rotated, upside down, or that do not load will not receive credit. Illegible or incomplete submissions will lose credit depending on what can be read. Ensure that your final submission contains all pages.

You are responsible for making sure your submission went through successfully.

Written submissions may be turned in up to one day late for a 50% penalty.

No grace day usage is allowed.

1 Proof by Induction

In this written assignment, you will use induction to prove the runtime of the recursive implementation of `fibonacci` shown below.

```
public int fibonacci(int n) {
  if (n <= 2) {
    return 1;
  } else {
    return fibonacci(n - 1) + fibonacci(n - 2);
  }
}
```

Setup and Hypothesis - 15 points

Before we prove anything about the runtime of the code above, we have to determine what the code's runtime growth function is, and come up with a hypothesis for its closed-form upper bound.

- **(5 pt)** Write out the recursive form of the growth function, $T(n)$ for `fibonacci`. Any constant costs can be written as $\Theta(1)$.
- **(5 pt)** Draw the recursion tree for `fibonacci`. Label each node with the runtime of a call to `fibonacci`, excluding the cost of recursive calls. Label the height of your tree in terms of n.
- **(5 pt)** Based on your recursion tree, give a hypothesis for the closed-form, tight upper bound of your growth function (i.e., a hypothesis of the form $T(n) \in O(f(n))$). In at most two sentences (or a summation), explain how you used your recursion tree to come up with the hypothesis.

Base Case - 10 points

- **(10 pt)** Prove that your hypothesis holds true for an appropriate number of base cases. Make sure to consider how many base cases you will need to prove, based on your growth function.

Inductive Case - 20 points

In order to prove that our hypothesis is true for **all** values of n, we must use induction. Remember that this works by proving that $P(n') \implies P(n)$ for some $n' < n$, where $P(n)$ is your hypothesis parameterized over n.

- **(5 pt)** Make an appropriate inductive assumption. Note, your assumption should make it possible to show that your hypothesis holds true for n when the assumption is true. You may need to make an assumption about more than just one smaller value of n'.
- **(15 pt)** Prove that if your inductive assumption is true, then your hypothesis must be true for $T(n)$.

Followup - 5 points

Just because a recursive solution works does not mean it is the only (or even the most efficient) choice...

- **(5 pt)** Give an iterative implementation for `fibonacci` and state the tigh big-O bound on the runtime.