0 Overview

Instructions

Due Date: Sunday, Mar 10 @ 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 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 submissions may also 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` show below.

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

Part 1 - Setup and Hypothesis

Before we prove anything about the runtime of the code above, we have to determine what the growth function is, and come up with a hypothesis for the runtime bounds.

1. [5 points] Write out the recursive form of the growth function, $T(n)$, for `fibonacci`.
2. [5 points] Draw the recursion tree for `fibonacci`. Label each node with the runtime of a call to `fibonacci` excluding the cost of the recursive calls. Label the height of your tree in terms of n.
3. [5 points] Based on your recursion tree, give a hypothesis for the closed-form tight upper bound of your growth function (ie a hypothesis of the form $T(n) \in O(f(n))$). In at most two sentences explain how you used your recursion tree to come up with the hypothesis.

Part 2 - Base Case

Now that you have a hypothesis, we must prove it for some number of base cases.

4. [10 points] 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.

Part 3 - Inductive Case

In order to prove that our hypothesis is true for ALL values of n, we must use induction.

5. [5 points] Make an appropriate inductive assumption.
6. [15 points] Prove that if your inductive assumption is true, then your hypothesis must be true for $T(n)$.

Part 4 - Followup

Just because a recursive solution works does not mean it is the most efficient choice...

7. [5 points] Give an iterative implementation for `fibonacci` and state the tight big-O bound on the runtime.