ICS 6D **Due: Wednesday, April 20, 2016, 3:00PM**

Homework 3

Instructor: Sandy Irani Sections 8.6 - 8.11

In each of the inductive proofs below, you must first state that you are proving the theorem by induction. The base case and inductive step must be clearly labeled. At the beginning of the inductive step, you need to state clearly what you are assuming and what you are proving. You must also clearly indicate where you are using the inductive hypothesis.

1. Prove that any amount of postage worth 24 cents or more can be made from 5-cent or 7-cent stamps.

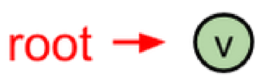
2. The sequence $\{g_n\}$ is defined recursively as follows:

   - $h_0 = \frac{5}{3}$
   - $h_1 = \frac{11}{3}$.
   - $h_n = 3 \cdot h_{n-1} + 4 \cdot h_{h-2} + 6n$, for $n \geq 2$.

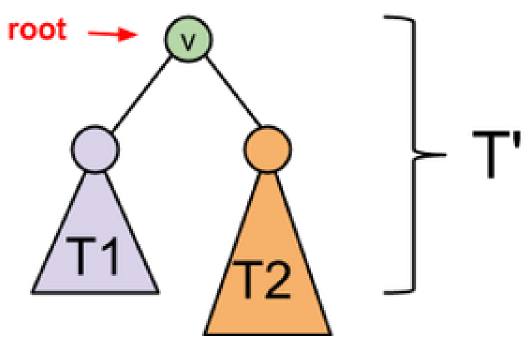
   Use induction to show that $h_n = 2 \cdot 4^n + \frac{3}{2}(-1)^n - n - \frac{11}{6}$.

3. Here is a definition for a set of trees called *full binary trees*.

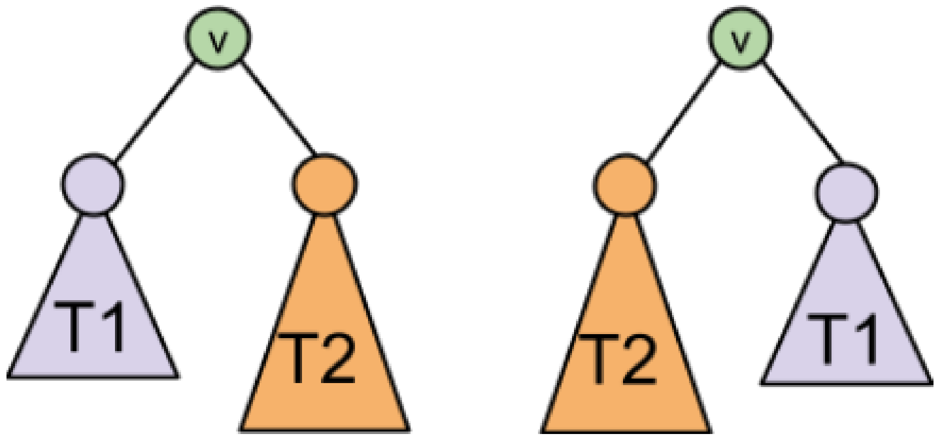
   Basis: A single vertex with no edges is a full binary tree. The root is the only vertex in the tree.



   Recursive rule: If T1 and T2 are full binary trees, then a new tree T' can be constructed by first placing T1 to the left of T2, adding a new vertex v at the top and then adding an edge between v and the root of T1 and an edge between v and the root of T2. The new vertex v is the root of T'.



   Note that it makes a difference which tree is placed on the left and which tree is placed on the right. The two trees below are considered to be different full binary trees:



(a) Draw all possible full binary trees with 3 or fewer vertices.

(b) Draw all possible full binary trees with 5 vertices.

(c) Draw all possible full binary trees with 7 vertices.

(d) Use structural induction to prove that every full binary tree has an odd number of vertices. (An integer x is odd if x = 2k + 1, for some integer k.)

4. Let $B = \{a, b\}$.

   (a) Give a recursive definition for B^*.

   (b) The set B^+ is the set of strings over the alphabet a, b of length at least 1. That is $B^+ = B^* - \{\lambda\}$. Give a recursive definition for B^+.

   (c) Give a recursive definition for the set S which is all strings from B^* such that all the a's come before all the b's. For example $aabbbb \in S$, but $aababbbb \notin S$.

5. Give a recursive algorithm that takes as input two positive integers x and y and returns the product of x and y. The only arithmetic operations your algorithm can perform are addition or subtraction. Furthermore, your algorithm should have no loops.

6. Give a recursive algorithm that takes as input a non-negative integer n and returns a set containing all binary strings of length n. Here are the operations on strings and sets you can use:

   - Initialize an empty set S.
   - Add a string x to S.
   - $y := 0x$ (This operation adds a 0 to the beginning of string x and assigns the result to string y).
   - $y := 1x$ (This operation adds a 1 to the beginning of string x and assigns the result to string y).
   - return(S)

   Also, you can have a looping structure that performs an operation on every string in a set:

```
For every x in S
     //perform some steps with string x
```

7. Give a recursive algorithm to compute the sum of the cubes of the first n positive integers. The input to the algorithm is a positive integer n. The output is $\sum_{j=1}^{n} j^3$. The algorithm should be recursive, it should not compute the sum using a closed form expression or a loop.

8. Use induction to prove that your algorithm in the previous question returns the correct result.
9. Give a recursive algorithm whose input is a, a real number and integer n such that $n \geq 0$, and whose output is a^{2^n}. **Note that the exponent of a is 2^n.**
10. Use induction to prove that your algorithm in the previous question returns the correct result.
11. Give the characteristic equation for the following recurrence relations:
    (a) $a_n = 3a_{n-1} - a_{n-2} + 17a_{n-3}$.
    (b) $a_n = -a_{n-2} + a_{n-4}$
    (c) $a_n = 3a_{n-7}$
12. Each equation below is a characteristic equation for a recurrence relation. Express the solution to each recurrence relation as a linear combination of terms. Note that you will not know the actual coefficients in the linear combination since you are not given the base cases, so you can use a_1, a_2, etc.
    (a) $(x-1)(x+2) = 0$
    (b) $(x-4)(x-4) = 0$
    (c) $(x-4)(x-4)(x+5) = 0$
    (d) $(x-4)(x-4)(x+5)^4 = 0$
13. Solve the following recurrence relation:
    - $f_n = f_{n-1} + 12f_{n-2}$
    - $f_0 = -2$
    - $f_1 = 20$
14. Solve the following recurrence relation:
    - $f_n = 3f_{n-1} + 4f_{n-2}$
    - $f_0 = 4$
    - $f_1 = 1$
15. Solve the following recurrence relation:
    - $f_n = 4f_{n-1} - 4f_{n-2}$
    - $f_0 = 3$
    - $f_1 = -10$

The following problems are for practice only. They will not be graded.

1. - $f_0 = \frac{5}{3}$
   - $f_1 = \frac{11}{3}$.
   - $f_n = 3 \cdot f_{n-1} + 4 \cdot f_{n-2} + 6n$, for $n \geq 2$.

   Use induction to show that $f_n = 2 \cdot 4^n + \frac{3}{2}(-1)^n - n - \frac{11}{6}$.
2. Solve the following recurrence relation:
   - $f_n = 4f_{n-2}$
   - $f_0 = 2$

- $f_1 = 16$

3. Solve the following recurrence relation:
   - $f_n = 3f_{n-1} + 4f_{n-2} - 12f_{n-3}$
   - $f_0 = 4$
   - $f_1 = -5$
   - $f_2 = 11$

4. Solve the following recurrence relation:
   - $f_n = 5f_{n-1} - 8f_{n-2} + 4f_{n-3}$
   - $f_0 = 6$
   - $f_1 = 7$
   - $f_2 = 17$