CSE 311: Foundations of Computing (Fall, 2015)

Homework 6 Out: Fri, 13-Nov. Due: Friday, 20-Nov, **11: 59 pm** on **Gradescope**

Additional directions: You should write down carefully argued solutions to the following problems. Your first goal is to be complete and correct. A secondary goal is to keep your answers simple and succinct. The idea is that your solution should be easy to understand for someone who has just seen the problem for the first time. (Re-read your answers with this standard in mind!) You may use any results proved in lecture (without proof). Anything else must be argued rigorously. Make sure you indicate the specific steps of your proofs by induction.

1. Non-crossing family of sets (20 points)

We say two sets $A, B \subseteq \{1,2,\ldots,n\}$ are *crossing* when $A \cap B$, $A \setminus B$, and $B \setminus A$ are all non-empty; we say A, B are non-crossing otherwise. Recall that $[n] = \{1,2,\ldots,n\}$. Let $S \subseteq \mathcal{P}([n])$ be a family of subsets of $[n]$.

a) Prove that if any two sets $A, B \in S$ are non-crossing, then $|S| \leq 2n$.

b) Now, we introduce a slightly different notion of non-crossing. We say, A, B are *properly crossing* when $A \cap B$, $A \setminus B$, $B \setminus A$, and $([n] \setminus A) \setminus B$ are all non-empty, and we say A, B are *properly non-crossing* otherwise.

   Prove that if any two sets $A, B \in S$ are properly non-crossing, then $|S| \leq 4n$.

2. Not this again (15 points)

Suppose the word size on our computer is large enough so that the number M fits into a single register. Consider the following pseudo-code that computes $f_n \bmod M$.

Fib(n):

If $n == 0$,

return 0;

else if ($n == 1$)

return 1;

else

return $(\mathrm{Fib}(n-1) + \mathrm{Fib}(n-2)) \bmod M$;

Let $R(n)$ be the number of additions that occur when Fib(n) is called. Using strong induction, prove that for any $n \geq 1$,

$$1.6^{n-1} - 1 \leq R(n) \leq 1.7^n.$$

3. Recursive definition of sets (15 points)

Let S be the set defined as follows.

Basis: $5 \in S$; $8 \in S$;

Recursive: if $x, y \in S$, then $x + y \in S$.

Show that there is an integer a such that for every integer $n \geq a$, it holds that $n \in S$.

4. Constructing sets (15 points)

For each of the following, write a recursive definition of sets satisfying the following properties. Briefly justify that your solution is correct.

(a) Binary strings where every occurrence of a 1 is followed by a 0.

(b) Binary string with even number of 1s.

(c) Binary strings with equal number of 0s and 1s.

5. Binary Search Trees (25 points)

Consider the following recursive definition of a binary tree:

$\circ$ is a tree

If L and R are trees, then $\mathrm{Tree}(x, L, R)$ is a tree.

We recursively define a function sorted which tests if a tree is in sorted order:

$$\begin{aligned}
&\mathrm{sorted}(\circ) &&= \mathrm{True} \\
&\mathrm{sorted}\big(\mathrm{Tree}(x,\circ,\circ)\big) &&= \mathrm{True} \\
&\mathrm{sorted}\big(\mathrm{Tree}(x,\circ,\mathrm{Tree}(y, L, R))\big) &&= \mathrm{sorted}\big(\mathrm{Tree}(x,\circ, L)\big) \wedge (x < y) \wedge \mathrm{sorted}\big(\mathrm{Tree}(y, L, R)\big) \\
&\mathrm{sorted}(\mathrm{Tree}(x, \mathrm{Tree}(y, L, R),\circ)) &&= \mathrm{sorted}\big(\mathrm{Tree}(y, L, R)\big) \wedge (y < x) \wedge \mathrm{sorted}\big(\mathrm{Tree}(x, R,\circ)\big) \\
&\mathrm{sorted}\big(\mathrm{Tree}(x, L, R)\big) &&= \mathrm{sorted}\big(\mathrm{Tree}(x, L,\circ)\big) \wedge \mathrm{sorted}\big(\mathrm{Tree}(x,\circ, R)\big)
\end{aligned}$$

Problem:

a) Suppose we have a binary tree $T = \mathrm{Tree}(x, L, R)$ such that $\mathrm{sorted}(T) = \mathrm{True}$. Use structural induction to show that for every node $y \in L$, $y < x$ and for every node $y \in R$, $x < y$.

b) Suppose $\mathrm{sorted}(T) = \mathrm{True}$. Define a recursive function $\mathrm{Find}(T, x)$ that returns True if T has a node with label x and returns False otherwise. Your algorithm is supposed to use at most $5 \cdot (1 + \mathrm{height}(T))$ comparisons to produce the output.

For an example, here is a recursive function insert which inserts a new value y into a sorted tree such that the new tree is also sorted.

$$\begin{aligned}
&\mathrm{insert}(\circ, y) &&= \mathrm{Tree}(y,\circ,\circ) \\
&\mathrm{insert}(\mathrm{Tree}(x, L, R), y) &&= \begin{cases} \mathrm{Tree}(x, L, R) & \text{if } y = x \\ \mathrm{Tree}(x, \mathrm{insert}(L, y), R) & \text{if } y < x \\ \mathrm{Tree}\big(x, L, \mathrm{insert}(R, y)\big) & \text{if } y > x \end{cases}
\end{aligned}$$

c) Use structural induction to prove that your function always returns the correct value and uses at most $5 \cdot (1 + \mathrm{height}(T))$ comparisons.