University of Washington
Department of Computer Science and Engineering
CSE 311, Spring 2013

May 15, 2013

# Homework 6, Due Wednesday, May 22, 2013

**Problem 1:**

The following recursive definition describes the running time needed by a recursive algorithm. Let $c \geq 0$ be an integer. Suppose that a function $T$ mapping integers $n \geq 0$ to integers satisfies

- $T(0) = 0$, $T(n) \leq c$ for $n \leq 20$,
- $T(n+1) \leq T(\lfloor n/5 \rfloor) + T(\lfloor 3n/4 \rfloor) + cn$ for $n \geq 20$.

Prove by (strong) induction that $T(n) \leq 20cn$ for all $n \geq 0$. (Hint: All you need about $\lfloor\ \rfloor$ is that for $x \geq 0$, $\lfloor x \rfloor$ is an integer and $0 \leq \lfloor x \rfloor \leq x$.)

**Problem 2:**

Consider the following one-player game: The player starts with an integer $n \geq 1$.
If $n = 1$ the game stops and the player has not earned any points.
If $n > 1$ the player gets to split $n$ into two positive integers $r$ and $n - r$. For this move, the player earns $r \cdot (n - r)$ points. After this, the player plays the game both on $r$ and on $n - r$, adding the points earned from those games to the points already earned.

Use induction to prove that no matter how the player plays on input $n \geq 1$, the player earns exactly $n(n-1)/2$ points.

**Problem 3:**

The set of *almost balanced* binary trees is a subset of all rooted binary trees and is defined in the same way as rooted binary trees except that the recursive step has an extra restriction:
In an almost balanced binary tree, one can only join trees $T_1$ and $T_2$ as in the rooted binary tree definition if either $\textbf{height}(T_1) = \textbf{height}(T_2)$ or $\textbf{height}(T_1)$ and $\textbf{height}(T_2)$ differ by 1. The functions **size** and **height** are defined exactly as for rooted binary trees.

Prove by induction that every almost balanced binary tree $T$ satisfies $\textbf{size}(T) \geq f_{\textbf{height}(T)+1}$ where $f_m$ denotes the $m$-th Fibonacci number. (As usual $f_0 = 0$, $f_1 = 1$, and $f_{m+1} = f_m + f_{m-1}$ for $m \geq 1$.)

**Problem 4:**

Construct regular expressions that match (generate) each of the following sets of strings.

a) The set of all binary strings that end with 0 and have even length, or start with 1 and have odd length.

b) The set of all binary strings that have a 1 in every even-numbered position counting from the start of the string with the start of the string counting as position 1.

**Problem 5:**

Construct regular expressions that match (generate) each of the following sets of strings.

a) The set of all binary strings that contain at least two 1's and at most two 0's.

b) The set of all binary strings that don't contain 001.

**Problem 6:**

Construct context-free grammars that generate the following sets of strings. For each of your constructions write a sentence or two to explain why your construction is correct. If you use more than one variable, as documentation explain what sets of strings you expect each variable to generate.

a) The set of all binary strings that contain at least two 0's and at most one 1.

b) The set of all binary strings that are of odd length and have 1 as their middle character.

**Problem 7:**

If $a \in \Sigma$ is a symbol then the string $a^n$ for $n \geq 0$ is the string consisting of $n$ copies of $a$, one after another. Construct context-free grammars that generate the following sets of strings. For each of your constructions write a sentence or two to explain why your construction is correct. If you use more than one variable, as documentation explain what sets of strings you expect each variable to generate.

a) $\{1^m 0^n 1^{m+n} : m, n \geq 0\}$.

b) $\{0^m 1^n 0^p : m, n, p \geq 0 \text{ and } m \geq n \text{ or } n \leq p\}$.

**Problem 8:**

Define a grammar by $S \to SS \mid 0S1 \mid 1S0 \mid \lambda$. Use structural induction to prove that every string generated by $S$ has an equal number of 0's and 1's.

**Extra Credit 9:**

For the grammar given in problem 8, use ordinary induction to prove that for every integer $n \geq 0$, every binary string of length $n$ with an equal number of 0's and 1's can be generated from $S$.

**Extra Credit 10:**

Consider the set $S_3$ of strings in $\{0,1,2\}^*$ such that the sum of the values is congruent to 0 modulo 3, so

$$S_3 = \{\lambda, 0, 00, 12, 21, 000, 012, 021, 102, 111, 120, 201, 210, 222, \cdots\}.$$

a) Design a context-free grammar that generates $S_3$.

b) Design a regular expression that generates $S_3$.