Computer Science CSC 238H | Wednesday July 8 1998
St. George Campus | University of Toronto

Homework Assignment #3
Due: Wednesday August 5 1998 at 6:10pm

Don't forget to attach the completed COVER PAGE *to the* front *of your assignment!*

1. [18 marks] Suppose you are given a binary search tree (BST) whose vertices are labelled by distinct positive integers. (BST's were defined in the lectures.) You would like to print out every integer x for which x appears as the label of some vertex but $x-1$ does not.

Consider the following recursive algorithm.

```
foo(v, p): integer
% preconditions: v is the root of a non-empty BST with vertices labelled by distinct integers
% and p is an integer smaller than the label of any vertex in the BST rooted at v.
    if v has a left child then p' ← foo(left child of v, p) else p' ← p
    if p' ≠ (label of v) − 1 then print the label of v
    largest ← label of v
    if v has a right child then largest ← foo(right child of v, label of v)
    return largest
end foo
```

[15] **(a)** Prove that if the procedure foo is called (with inputs satisfying the preconditions),
(i) it terminates,
(ii) it returns the largest label in the BST rooted at v, and
(iii) it prints those integers x such that (a) x appears in the BST rooted at v and (b) $x-1$ does not appear in the BST rooted at v and (c) $x-1 \neq p$.

(Thus, if the algorithm is called with v equal to the root of the original BST and p equal to -1, it will solve the task that you set out to solve.)

[3] **(b)** Let n be the number of vertices in the BST rooted at v. Give a tight bound on the running time of the algorithm in terms of n using big-Θ notation. Explain why your bound is correct.

2. [8 marks] Solve the following recurrence:

$$\begin{aligned} f(0) &= 1 \\ f(1) &= 7 \\ f(n) &= f(n-2)+1 \text{ for } n>1. \end{aligned}$$

For part marks, you may just solve the recurrence for even values of n.

3. [20 marks] When using the "guess-and-check" method of solving recurrences it is often hard to spot the pattern. In this question we shall look at a more systematic way of coming up with our guess.

We shall solve the following recurrence.

$$\begin{aligned} b(0) &= 3 \\ b(1) &= 1 \\ b(n) &= b(n-1) + 2b(n-2) + 6 \text{ for } n \geq 2. \end{aligned}$$

We shall first define a new function $\mathcal{B}(x)$, where x is a real number, by using the values of b as coefficients of a power series:

$$\mathcal{B}(x) = \sum_{n=0}^{\infty} b(n)x^n = b(0)x^0 + b(1)x^1 + b(2)x^2 + b(3)x^3 + \cdots.$$

(For those of you who are familiar with infinite series, do not worry about whether this series converges; all of this can be made rigorous, but for now, we are just using this technique to arrive at a guess for the solution of the recurrence, which will then be proved correct in the usual (rigorous) way. In fact, it is not hard to prove that the series is absolutely convergent for $|x| < \frac{1}{3}$ using the ratio test from first-year calculus. We assume throughout this question that $|x| < \frac{1}{3}$.)

In part (a), you will use the recurrence to find an equation involving $\mathcal{B}(x)$. In part (b), you will solve the equation to get an explicit formula for $\mathcal{B}(x)$. In parts (c) and (d), you will expand the formula found in part (b) into the sum of two geometric series. In part (e), you will use these series to guess a solution to the original recurrence.

[4] **(a)** Use the recurrence to show that

$$\begin{aligned} \mathcal{B}(x) &= 3 + x + \sum_{n=2}^{\infty} b(n)x^n \\ &= 3 + x + x(\mathcal{B}(x) - 3) + 2x^2\mathcal{B}(x) + \sum_{n=2}^{\infty}(6x^n) \end{aligned}$$

[3] **(b)** Solve the above equation for $\mathcal{B}(x)$ to show that

$$\mathcal{B}(x) = \frac{8x^2 - 5x + 3}{(1-x)(1-2x)(1+x)}.$$

You probably learned the following fact about the sum of an infinite geometric series in high school: $\sum_{n=0}^{\infty} az^n = \frac{a}{1-z}$ (for $|z| < 1$). You may use this fact here without proving it.

[3] **(c)** Find three real numbers E, F and G so that

$$\mathcal{B}(x) = \frac{E}{1-x} + \frac{F}{1-2x} + \frac{G}{1+x}.$$

[3] **(d)** Use the fact about geometric series mentioned in part (b) to show that

$$\mathcal{B}(x) = \sum_{n=0}^{\infty}(E + F \cdot 2^n + G \cdot (-1)^n)x^n,$$

where E, F and G are the numbers you found in part (c).

[7] **(e)** Using part (d) and the original definition of the function $\mathcal{B}(x)$, guess a solution to the recurrence, and prove that it is correct by induction on n.

4. [16 marks] Suppose we have two subroutines A and B that call themselves recursively and also call each other. This type of programming is called mutual recursion.

When given an input of size $n \geq 2$, routine A calls itself once on an input of size $n-1$ and routine B four times on inputs of size $n-1$ and then does one extra step. When given an input of size $n \geq 2$, routine B calls itself twice and calls A three times and each subroutine call has an input of size $n-1$. Let $a(n)$ be the running time of A on inputs of size n. Let $b(n)$ be the running time of B on inputs of size n. Suppose

$$a(1) = 17 \text{ and } b(1) = 13.$$

[4] **(a)** Write down recurrence relations that define $a(n)$ and $b(n)$ for $n \geq 2$.

[2] **(b)** Use the recurrence relations to compute the values of $a(n)$ and $b(n)$ for $n \in \{1, 2, 3, 4, 5\}$.

[10] **(c)** Prove that the running times of both procedures A and B are $\Theta(5^n)$. Notice that you need not explicitly solve the recurrences! Hint: Look at the proof on page 182 of the text, which establishes a bound on a more complicated recurrence without explicitly solving it. You will get a few part marks if you just prove that $a(n)$ and $b(n)$ are both $\Omega(5^n)$.

5. [8 marks] Design a deterministic finite automaton that accepts the language
$L = \{x \in \{0,1\}^* \mid$ in *every* prefix of x, the number of 1's differs from the number of 0's by at most $1\}$.
The string y is called a prefix of x if there is some string z such that $yz = x$. The string 0101010101 is in the language L since, in any prefix, the number of 0's differs from the number of 1's by 0 or 1. The string 00011011 is not in the language, since the prefix 000 has three 0's but no 1's. The string 100111101000 is not in L since the prefix 100111 has four 1's and only two 0's. Try to make your automaton as simple as possible. Briefly explain why your automaton accepts the language L. (You do not have to give a formal proof.)

6. [15 marks] Consider the following non-deterministic finite automaton A with alphabet $\{0,1\}$.

[5] **(a)** Write down an English description of the strings that are accepted by A.

[5] **(b)** Write down a simple regular expression for the language accepted by A. Briefly explain why your answer is correct.

[5] **(c)** Draw a deterministic finite automaton that recognizes the same language as A does. Briefly explain why your answer is correct.

7. [15 marks] Let L be a language over $\{0,1\}^*$. Let

$$\text{INIT}(L) = \{x \in \{0,1\}^* | x \text{ is a prefix of some string } y \in L\}.$$

Prove that if L is regular, so is $\text{INIT}(L)$. Hint: Modify a DFA that accepts L to create a DFA that accepts $\text{INIT}(L)$.