Homework 11: Due Wednesday, April 12

Problem 1: In class, we defined full binary trees using the following datatype:

$$\text{datatype tree} = \text{Leaf of int} \mid \text{Internal of } (\text{int} * \text{tree} * \text{tree}).$$

We also defined the function

$$\begin{aligned}\text{fun } & \text{numNodes}(\text{Leaf}(a)) = 1 \\ & \mid \text{numNodes}(\text{Internal}(a, s, t)) = \text{numNodes}(s) + \text{numNodes}(t) + 1.\end{aligned}$$

Show directly by structural induction that numNodes(t) is odd for all full binary trees t. Use only the definition of odd and the above definition of numNodes (so do not use numLinks, or any other result).

Interlude: Suppose that we want to define more general binary trees (not just "full" ones). One approach is to replace Internal in the above definition with three possibilities, Left, Right, and Both, representing when a node has only a left child, only a right child, or both children:

$$\text{datatype tree} = \text{Leaf of int} \mid \text{Left of } (\text{int} * \text{tree}) \mid \text{Right of } (\text{int} * \text{tree}) \mid \text{Both of } (\text{int} * \text{tree} * \text{tree}).$$

Alternatively, we can introduce a dead end marker, and then just have general nodes, which leads to the following definition:

$$\text{datatype binTree} = \text{Null} \mid \text{Node of } (\text{int} * \text{binTree} * \text{binTree}).$$

In other words, we start with the empty tree, and allow one side below a node to be Null while the other continues to grow. For example,

$$\text{Node}(1, \text{Null}, \text{Node}(2, \text{Null}, \text{Null}))$$

as a binary tree, as is

$$\text{Node}(1, \text{Node}(4, \text{Null}, \text{Null}), \text{Node}(2, \text{Node}(3, \text{Null}, \text{Null}), \text{Null})).$$

Use this definition of binTree in Problems 2 through 4 below.

Problem 2: Working with the binTree datatype, write the following ML functions:
a. A function numNulls that takes a binary tree as input, and produces the integer that is the number of Nulls in the tree.
b. A function numNodes that takes a binary tree as input, and produces the integer that is the number of Nodes in the tree.

Problem 3: Using your definitions of numNulls and numNodes in Problem 2, prove the following by structural induction on binary trees: For all binary trees t, we have numNulls(t) = numNodes(t) + 1.

Problem 4:
a. Write an ML function binTreeMap that takes two inputs, a function f of type int $\rightarrow$ int and a binary tree t, and produces the binary tree that results from applying the function f to each integer in each node of t.
b. Write an ML function flattenBinTree that takes a binary tree as input, and returns the list of all integers in order when read across the tree from left to right. For example, on the input

$$\text{Node}(1, \text{Node}(4, \text{Null}, \text{Null}), \text{Node}(2, \text{Node}(3, \text{Null}, \text{Null}), \text{Null})),$$

the program should produce

$$[4, 1, 3, 2],$$

and on the input

$$\mathsf{Node}(7, \mathsf{Node}(4, \mathsf{Node}(2, \mathsf{Node}(1, \mathsf{Null}, \mathsf{Null}), \mathsf{Node}(3, \mathsf{Null}, \mathsf{Null})), \mathsf{Node}(5, \mathsf{Null}, \mathsf{Null})), \mathsf{Node}(11, \mathsf{Null}, \mathsf{Null})),$$

the program should produce

$$[\mathsf{1}, \mathsf{2}, \mathsf{3}, \mathsf{4}, \mathsf{5}, \mathsf{7}, \mathsf{11}].$$

Problem 5: Show that

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

for all $n \in \mathbb{N}^+$.