Lab 04

15-150 Fall 2024

Due: Thu 19th Sept, 2024 at 5:00pm

Interview

This assignment will be evaluated in an interview with the course staff on the date above. The points in the tasks are meaningless and submission of solutions is not required.

The instructions for the interview are in the interview guide.

1 Work It Out

Consider the following two functions:

```
fun listMax ([] : int list) : int = 0
  | listMax (x::xs) = Int.max (x, listMax xs)

fun treeMax (Empty : tree) : int = 0
  | treeMax (Node (l,x,r)) =
      Int.max (treeMax l, Int.max (x, treeMax r))
```

For each of the functions above:

- write the function's recurrence relations;
- solve the recurrence using the tree or telescopic sum method;
- derive a tight big-O bound;
- briefly explain why your answer is correct.

You may not use the table of common recurrences for these problems. The goal is to practice solving recurrences.

Task 1.1 (5 points) Find the *work* of `listMax`, in terms of the length, n.

Task 1.2 (5 points) Find the *span* of `listMax`, in terms of the length, n.

1.1 A balanced tree

For the following tasks, assume the tree is balanced.

Task 1.3 (5 points) Find the *work* of `treeMax`, in terms of the number of nodes in the tree, n.

Task 1.4 (5 points) Find the *span* of `treeMax`, in terms of the number of nodes in the tree, n.

Task 1.5 (5 points) Find the *work* of `treeMax`, in terms of the tree's depth, d.

Task 1.6 (5 points) Find the *span* of `treeMax`, in terms of the tree's depth, d.

1.2 An unbalanced tree

For the following tasks, assume the tree is maximally unbalanced.

Task 1.7 (5 points) Find the *work* of `treeMax`, in terms of the number of nodes in the tree, n.

Task 1.8 (5 points) You should have gotten the same asymptotic bound for the work of `treeMax` in both the balanced case and the maximally unbalanced case. It turns out that the work of `treeMax` only depends on n, the number of nodes in the tree, and not the structure of the tree. Why is this the case? (A brief justification suffices.)

Hint: Suppose there are n_l nodes in the left subtree and n_r nodes in the right subtree. Write the recurrence for this setup.

Task 1.9 (5 points) Find the *span* of `treeMax`, in terms of the number of nodes in the tree, n.

Task 1.10 (5 points) Find the *work* of `treeMax`, in terms of the tree's depth, d.

Task 1.11 (5 points) Find the *span* of `treeMax`, in terms of the tree's depth, d.

1.3 Comparing trees

Task 1.12 (5 points) Compare your answers for the balanced and unbalanced cases. Which kind of tree (balanced or maximally unbalanced) is the worst case for `treeMax` when analyzed in terms of depth?

How can we determine the worst case for `treeMax` without doing both analyses?