Data Structures

Fall 2020, Midterm Exam.

Nov. 9, 2020

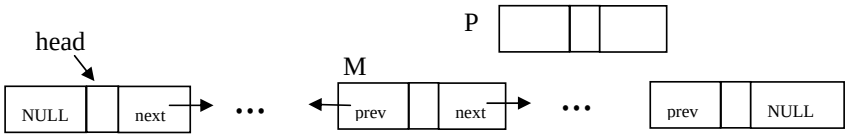
1. (10 pts) List the following functions by increasing asymptotic growth rate. If two functions have the same asymptotic growth rate, state that fact by circling them. No justification is needed. Here $\log n$ stands for $\log_2 n$.

$$\sqrt{n} \quad (\log n)^3 \quad 1.1^n \quad n^2 \log n \quad n(\log n)^2 \quad 3\log n \quad 2^5 \quad (100n)^{0.1} \quad \frac{1}{n} \quad (\frac{n}{100})^{1.5}$$

Sol.

$$\frac{1}{n} \quad 2^5 \quad 3\log n \quad (\log n)^3 \quad (100n)^{0.1} \quad \sqrt{n} \quad n(\log n)^2 \quad (\frac{n}{100})^{1.5} \quad n^2 \log n \quad 1.1^n$$

2. (10 pts) Assume that LL is a DOUBLY linked list with *head* pointing to the first node and at least one other internal node M which is not the last node. Write few lines of code to accomplish the following two operations. You may assume that each node has a *next* pointer and *prev* pointer. Given a node X, $X.next$ (resp., $X.prev$) gives the address of the node following (resp., before) X. Note that $(X.next).prev$ is the address stored in the *prev* field of the node whose address is $X.next$, as long as X is not the last node. Let $NULL$ be the null pointer constant. Note that the content of the *prev* (resp., *next*) field for the first (resp., last) node is $NULL$ as the following figure shows. No need to allocate or free memory in your codes.



   (a) Delete the first node, i.e., the node pointed to by *head*.
   Sol.
   $head = head.next$
   $head.prev = NULL$

   (b) Insert a node P immediately after M.
   Sol.
   $P.next = M.next$
   $M.next = P$
   $(P.next).prev = P$
   $P.prev = M$

3. (10 pts) Consider the following recursive function foo

```
int foo(int a, int b)
    if (a%b == 0)
      return b
    else return foo(b, a%b)
```

   Recall that $a\%b$ is the remainder of a divided by b. For instance $13\%10 = 3$. $x == y$ is true if the values of x and y are equal. Answer the following questions:

   (a) (3 pts) What is the output given by foo(17, 3)?
   Sol. 1

   (b) (3 pts) What is the output given by foo(3, 9) ?
   Sol. 3

   (c) (4 pts) Explain briefly the purpose of the foo function.
   Sol. GCD

4. (10 pts) For each of the following scenarios choose the best data structure from the following list or a combination of data structures: an unsorted array, linked list (LL), doubly LL (DLL), circular LL, circular DLL, stack, queue, binary search tree, AVL tree, splay tree. No explanations needed.

(a) Suppose that a grocery store decided that customers who come first will be served first.
Sol Queue

(b) A list must be maintained so that any element can be accessed randomly.
Sol. Array

(c) A program needs to remember operations it performed in opposite order.
Sol. Stack

(d) A list must be maintained so that elements can be added to the beginning or end in O(1).
Sol. Circular DLL

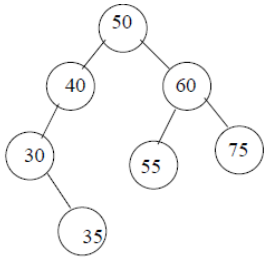
(e) Given a set of numerical data (such as 5, 100, 2 ...), a data element can be found in fewer than $O(n)$ steps in the worst case.
Sol. AVL Tree

5. (10 pts) Suppose the postorder traversal sequence of a binary search tree is

$$\mathbf{35}, \mathbf{30}, \mathbf{40}, \mathbf{55}, \mathbf{75}, \mathbf{60}, \mathbf{50}.$$

(a) Draw the tree.

(b) Is the tree unique? Briefly explain why.

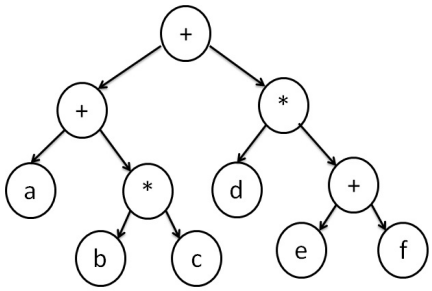


Sol: Yes, it is unique.

6. (10 pts) Consider the following postfix expression:

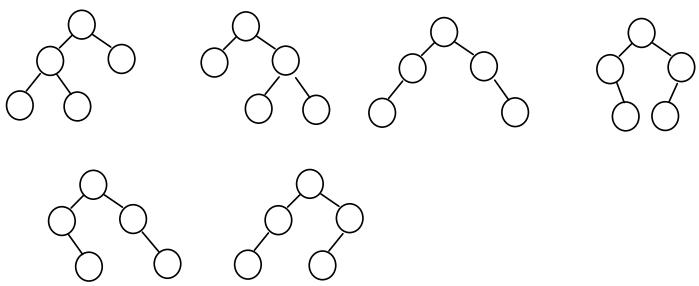
$$a \quad b \quad c \quad * \quad + \quad d \quad e \quad f \quad + \quad * \quad +$$

(a) What is the corresponding prefix expression?

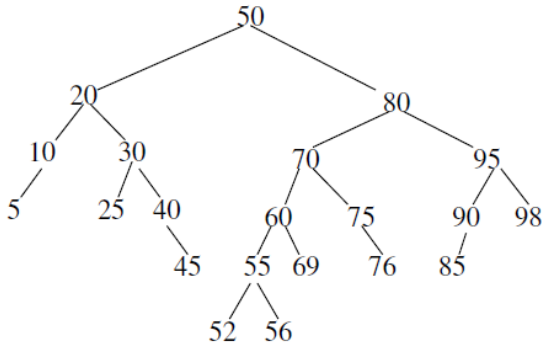


(b) Draw the expression tree associated with the postfix expression. Recall that an expression tree is a binary tree with internal nodes representing operators and leaves representing operands.
Sol: + + a * b c * d + e f

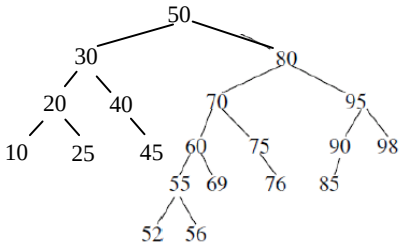
7. (10 pts) Draw the structures of all AVL-trees with 5 nodes.
Sol.

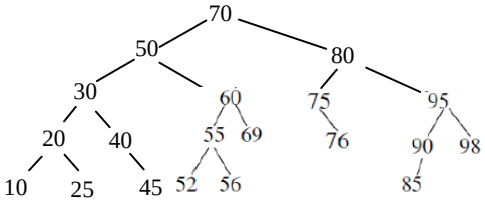


8. (10 pts) Delete element 5 in the following AVL tree and rebalance the tree. Show (i.e., draw) each of the intermediate trees after a single or a double-rotation is carried out.



Sol.



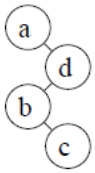


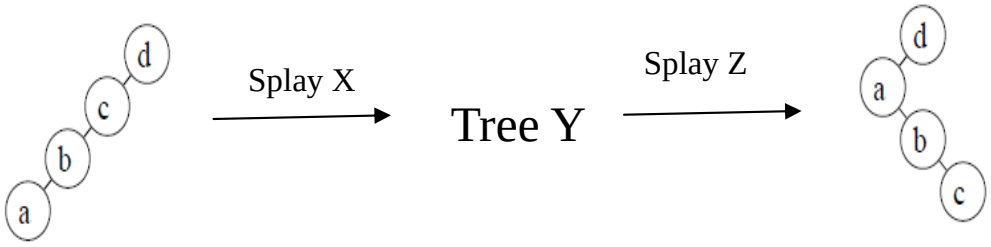
9. (10 pts) Given the tree on the left, suppose we want to perform two splay operations (based on bottom-up (i.e., 2-pass) splaying) on nodes X and Z ($X, Z \in \{a, b, c, d\}$) to turn the left tree into the right tree with Tree Y as the intermediate tree.

   (a) What are the two nodes X and Z?
   Sol.
   $X = a$, $Z = d$

   (b) What does the intermediate tree Y look like?

Sol.





10. (10 pts) Consider the splay tree data structure. Suppose you are given a set of keys $a_1, a_2, ..., a_n$ and an empty splay tree to start with.

   (a) How to sort the n keys (in increasing order) using the splay tree? Explain your method in Chinese or English. Do not write a program.
   Sol.
   Step 1: Insert the n keys into an initially empty splay tree.
   Step 2: Output the inorder traversal sequence.

(b) What is the worst case running time (in $O(...)$) of your sorting method? Why? Your answer for the running time should be as tight as possible.

Sol.

Step 1: $O(n \log n)$ time as each insertion of a splay tree takes $O(\log n)$ amortized time, and there are n such operation

Step 2: inorder traversal takes $O(n)$ time

Hence, the overall running time is $O(n \log n)$