COMP 2119C Introduction to Data Structures and Algorithms

Assignment Four

Due Date: 20 April 2023 23:55pm

[Note that for the questions involving algorithm design, before you present your algorithm, try to describe your idea first! More marks will be given to faster algorithms.]

1) (a) Insert the following numbers into an initially empty *AVL tree* one by one. Show the resulting tree after each insertion (including the “balanced factor” using the notation in the lecture slide). Then, delete 215, 238. Show also the resulting tree (with the balance factor) after each deletion.

157, 144, 131, 238, 176, 215, 217, 162

(b) Consider each of the following cases. If it is correct, give a proof for (i) and (ii); provide an algorithm to reconstruct the tree for (iii) and (iv). If it is incorrect, give a counter-example.

(i) In a binary search tree, if node *u* has two children, its predecessor has no left child.

(ii) In a binary search tree, if node *u* has one child, its predecessor has no right child.

(iii) Given the *preorder* and *postorder* traversal of a binary tree, we can uniquely determine the topology of the binary tree (i.e, there does NOT exist two different binary trees with the same preorder and postorder traversals).

(iv) Given a *preorder* traversal of a binary search tree, we can uniquely determine the topology of the tree.

2) Consider the *average case analysis* for binary search. Let *A*[1..*n*] be the given array of sorted distinct numbers and let n = 2*k* for some integer *k*. Consider a *successful* search (i.e., the number you want to search exists in the array *A* and there are only *n* possible cases for input).

1. How many case(s) requiring only *one* comparison to locate the given number? How many cases requiring exactly *two* comparisons to locate the given number?
2. How many case(s) requiring exactly *i* comparisons to locate the given number (1 ≤ *i* ≤ *k*).
3. Hence, or otherwise, compute the average case complexity for a successful search in binary search assuming that all *n* cases are equally likely

3) Sort the following numbers in increasing order using (a) bubble sort; (b) insertion sort; (c) selection sort; (d) merge sort; Show the resulting list of numbers after each round for (a) – (d).

276, 245, 210, 236, 280, 224, 255, 29, 226, 290, 214

4)

1. Is the assignment (1) too difficult; (2) too easy; (3) about right?
2. How many hours you spend on the assignment?
3. Less than 5 hours
4. 5 – 10 hours
5. 10 – 20 hours
6. More than 20 hours
7. [Self assessment] Do you consider yourself understand the topics of this assignment?

(1) Yes; (2) not 100% sure; (3) No

If your answer is (2) or (3), please elaborate (at least indicate which part you do not understand).

1. Other comments?

--- End of Assignment ---