**After lecture13 & lecture14**

Practice Problems (all taken from previous exams)

1. If you want to create in order-statistic tree (which needs the size of each subtree rooted at each node), from an already created red-black tree, you can:

a) perform a pre-order traversal of the order-statistic tree and sum the sizes of each subtree of a node and add one to get the size of each node (nodes with no children assigned size=1)

b) perform an in-order traversal of the order-statistic tree and sum the sizes of each subtree of a node and add one to get the size of each node (nodes with no children assigned size=1)

c) perform a post-order traversal of the order-statistic tree and sum the sizes of each subtree of a node and add one to get the size of each node (nodes with no children assigned size=1)

2. How does an augmented data structure differ from a traditional data structure?

a) Augmented data structures have an asymptotically higher memory overhead.

b) Augmented data structures worsen the asymptotic runtime of basic operations.

c) Augmented data structures offer additional operations or information.

d) Augmented data structures have a faster runtime complexity than the non-augmented data structure.

3. If a problem can be broken into subproblems which are reused several times, the problem has \_\_\_\_.

a) Overlapping subproblems

b) Optimal substructure

c) Memoization

d) Greedy

4. What is the space complexity of the dynamic programming implementation of the matrix chain problem?

a) O(1)

b) O(n)

c) O(n^2)

d) O(n^3)

5. Given an element x in an n-node order-statistic tree and a natural number i, how can we determine the ith successor of x in the linear order of the tree in O(lg n) time? So x is a key in the tree and we want to find the ith key after x in linear order.

A picture containing screenshot, text, circle, diagram

Description automatically generated

6. Suppose that the dimensions of the matrices A, B, C, and D are 8x5, 5x11, 11x6, and 6x9 respectively, and that we want to parenthesize the product ABCD in a way that minimizes the number of scalar multiplications. Find the "m" and "s" tables computed by MATRIX-CHAIN-ORDER to solve this problem and show the optimal parenthesization.

7. Let *R*(*i*, *j*) be the number of times that table entry *m*[*i*, *j*] is referenced while computing other table entries in a call of MATRIX-CHAIN-ORDER. Show that the total number of references for the entire table is

fig360_01