CS251 - Data Structures and Algorithms
Fall 2024

PSO 3, Week 4

Question 1

(Binary Tree)

(1) A full binary tree cannot have which of the following number of nodes?

A. 3
B. 7
C. 11
D. 12
E. 15

(2) Given the number of nodes $n = 7$, how many distinct shapes can a full binary tree have?

A. 3
B. 4
C. 5
D. 6
E. 7

(3) The number of leaf nodes is always greater than the number of internal nodes in a full binary tree.

A. True
B. False

(4) The number of leaf nodes is always greater than the number of internal nodes in a complete binary tree.

A. True
B. False

(5) Given the number of nodes in a full binary tree, the number of its leaf nodes is determined.

A. True
B. False

Question 2

(Stack and Queue)

Design a stack using two queues satisfying the following requirements

1. Pushing an element to the stack takes no more than $O(1)$ operations.
2. Popping from the stack takes no more than $O(1)$ operations if performed after a push.
3. Popping from the stack takes no more than $O(n)$ operations if performed after another pop, where n is the number of elements in the data structure.

Question 3

(Binary heap)

(1) If the binary heap is represented as an array, and the root is stored at index 0, where is the left child of the node at index $i = 23$ stored?

A. 45

B. 46

C. 47

D. 48

E. 49

(2) If the binary heap is represented as an array, and the root is stored at index 0, where is the parent of the node at index $i = 99$ stored?

A. 45

B. 46

C. 47

D. 48

E. 49

(3) If the binary heap is represented as an array of length $n = 99$, and the root is stored at index 0, where is the last non-leaf node stored?

A. 45

B. 46

C. 47

D. 48

E. 49

(4) If the binary heap is represented as an array of length $n = 99$, and you want to insert an element, how many different locations of the element are possible after insertion?

A. 5

B. 6

C. 7

D. 8

E. 9

Question 4

(Review)

(1) The big-O closed-form runtime expression $T(n)$ for the recurrence $T(n) = 3T(n/3) + n$ is (assume n is a power of 3 and $T(1) = 1$)

A. $O(n)$

B. $O(n \log n)$

C. $O(n^3 \log n)$

D. $O(\sqrt[3]{n} \log n)$

E. $O(n \sqrt[3]{\log n})$

‘

(2) Two algorithms are developed based on the following template

function $\mathcal{A}$(n : $\mathbb{Z}_{\geq 1}$ power of 2)
    if $n = 1$ **then**
        return 1
    end if
    ____
    return $\mathcal{A}$(n/2) + $\mathcal{A}$(n/2)
end function

The missing part requires $F(n)$ time in Algorithm $\mathcal{A}_1$, and requires $G(n)$ time in Algorithm $\mathcal{A}_2$, where $F(n)$ and $G(n)$ are two functions of n.

If $F(n) = \Theta(G(n))$, then $\mathcal{A}_1(n) = \Theta(\mathcal{A}_2(n))$.

The above statement is

A. True

B. False

C. Possibly true/ Possible false

(3) Consider a sorted circular doubly-linked list where the head element points to the smallest element in the list. What is the time complexity to find the largest element in the list?

A. $O(1)$

B. $O(\log n)$

C. $O(n)$

D. $O(n \log n)$

‘