Question Bank

Topic: Binary Search Tree (BST), Doubly Linked List, In-place Conversion, O(n) Time Complexity

Stars	Question	Years
	Given a binary search tree (BST), complete the following C++ function to convert the BST to a sorted (descending order) doubly linked list. The running time of the algorithm must be O(n) and you must do the conversion in place (destroying the BST structure) without using any additional data structure. typedef struct Node { int value; Node *left, *right; } Node; // root is the pointer to root Node of the BST // head and tail of the doubly link list are initially null void convertBST(Node* root, Node* &head, Node* &tail) {}	2023 (1x)

Topic: Stacks, Expression Conversion (Infix to Postfix), Algorithms

Stars	Question	Years
	An infix expression is an expression of the form “a operator b” (for example, a + b), i.e., an operator is in-between every pair of operands, and a postfix expression is an expression of the form “a b operator” (for example, ab+), i.e., every pair of operands is followed by an operator. Provide a pseudocode to convert an infix expression to an equivalent postfix expression using a stack. You can assume that the operands will be single letters and the operators will be limited to +,, , and /. Sample input and output: Input Output A+B C +D ABC +D+ ((A+B) C (D/E)) + F AB + CDE/ F+	2023 (1x)

Topic: Queues, Algorithm Design, Data Structure Manipulation (In-place)

Stars	Question	Years
	Write an algorithm to reverse the order of elements in a given queue without using any additional data structures.	2023 (1x)

Topic: Number Systems, Iteration, Algorithms (Non-recursive)

Stars	Question	Years
	Write a function that takes an integer n and prints the octal representation of n. [The function cannot be recursive]	2023 (1x)

Topic: Binary Search Tree (BST), Find Min/Max, Error Analysis, Debugging

Stars	Question	Years

	The findMin function above is trying to find the minimum value in a BST and the findMax function above is trying to find the maximum value in a BST. Do you think the findMin and the findMax function would work properly? If not then write the correct code. [Hint: Don’t overthink]
	int findMin(Node* root) {
	//left subtree
	root = root->left;
	while (true) {
	if(root->left != NULL){
	root = root->left;
	} else {
	return root->value;
	}
	}
	}
	int findMax(Node* root) {
	//right subtree
	root = root->right;
	while (true) {
	if(root->right == NULL){
	return root->value;
	}
	}
	}
		2023 (1x)

Topic: Fibonacci Sequence, Algorithm, Data Structure (Stack/Queue context for reversal)

Stars	Question	Years
	Your task is to write pseudocode for two functions. The first one prints the first n Fibonacci numbers. The second function prints the first n Fibonacci numbers in reverse order. (Constraints mentioned: For answering the following question, you can only use i) The enqueue and dequeue operations of the Queue ADT or/and ii) The push and pop operations of the Stack ADT)	2023 (1x)

Topic: Linked List, Intersection, Sorted Lists, Algorithm

Stars	Question	Years
	Given two lists sorted in increasing order, write an algorithm to return a new list representing the intersection of the two lists. The new list should be made with its own memory, the original lists should not be changed. Example: Input: First linked list: 1->2->3->4->6 Second linked list be 2->4->6->8, Output: 2->4->6.	2023 (1x)

Topic: Singly Linked List, Fast/Slow Pointers, Algorithm Design

Stars	Question	Years

	Write a function getNodeValue(intfraction) that returns the element of the node in a singly linked list that is the mth node of the list. Assuming that there are n > 1 nodes, then mth node (we count from 1) is such that m = n/fraction. This function should never be called when the list is empty. You cannot use any variable to count the nodes or any other function in your implementation. Hint: Suppose Leonard runs twice as fast as Sheldon.Both started to sprint at the same time from the start. So, when Leonard reaches the finish line, where will Sheldon be? What will be the scenario if Leonard runs thrice as fast as Sheldon?	2023 (1x)