Roll No. [redacted] Section BSSE 3A

National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code:	CS 201
Program:	BS(CS)	Semester:	Fall 2021
Duration:	15 Minutes	Total Marks:	10
Paper Date:	9 Nov 2021	Exam	Quiz 2
Section:	A		

9/10

Instruction/Notes: Solve the exam on this question paper.

Assume you are given a Doubly link list class

Question1(10 marks): Write a recursive function to DLL class that returns true if the doubly link list is a palindrome. Think carefully about the input parameter of your function *IsDLLPalindrome*

```
void Wrapper ()
{   Node<T>* Ptr1 = Head;
    Node<T>* Ptr2 = Head;
    if ( isDLLPalindrome(Ptr1, Ptr2) == true)
        cout << " Palindrome ";
}
bool isDLLPalindrome ( Node<T>*& Ptr1, Node<T>* Ptr2)
{   if ( Ptr2 == 0)
        return true;
    else if (Ptr2)
    else
    return isDLLPalindrome (Ptr1 , Ptr2->next);

    if (Ptr1->data == Ptr2->data)
    return isDLLPalindrome(Ptr1->next,
                           Ptr2->previous);
    else
        return False;
}
```

This part won't execute because of return statement above

Roll No. _____ Section BS SE 3A

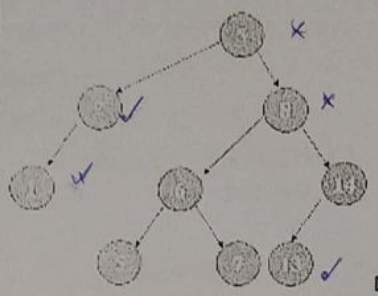
National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code:	CS 201
Program:	BS(SE)	Semester:	Fall 2021
Duration:	15 Minutes	Total Marks:	10
Paper Date:	30 Nov 2021	Exam	Quiz 3
Section:	A		

5/10

Instruction/Notes: Solve the exam on this question paper.

Question: Write a C++ function in a Binary Search Tree class that counts the number of nodes with one child at different depths of the binary tree and returns it.



Depth 3

Output

No of nodes with one child on
Depth 0 = 0
Depth 1 = 1
Depth 2 = 2
Depth 3 = 0

not checking properly

```
int Count ( Node<T>* curr , int depthNodes )
{   if (curr==0)
        return 0;
    else if (curr->left!=0 && curr->right==0)
    { return 1 + count(curr->left, depthNodes+1)
    else if (curr->right!=0 && curr->left==0)
    { return 1 + Count (curr->right);
      cout << " Depth : " depth+1 << "nodes="
           << depthNodes ;
    }
}
cout<< depthNodes;
}

void Count ()
{   cout << " Nodes = ";
    cout << Count (root->left, 0) + Count (root->right, 0);
}
```

4/10

Roll No. [redacted] Section BS SE 3A

National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code:	CS 201
Program:	BS(SE)	Semester:	Fall 2021
Duration:	10 Minutes	Total Marks:	10
Paper Date:	28 Sep 2021	Exam	Quiz 1
Section:	A		

Instruction/Notes: Solve the exam on this question paper.

Question: Consider the following program

i) **Give an estimate of T(N).** *(Show your work and give a T(N) estimate for each line of code.)*

```
sum = 0;                          O(1)
for( i = n; i > 0; i-- ){
        for( j = i; j > 0; j=j/2 )
                sum++;

        if(j%2==0)
           for( j = 1; j < i * i; j++ )
                ++sum;
}
```

T(N) for each line

→ O(1)

O(1) + O(N) + O(N)

O(N lg₂N) + O(N lg₂N) + O(N lg₂N)

→ O(N lg₂N)

→ O(N lg₂N) ∝

→ O(N lg₂N) + O(N² lg₂N) + O(N² lg₂N)

→ O(N² lg₂N) ∝

T(N) = 2 + 2N + 6N lg₂N + 3N² lg₂N

9

i	j	#	k
5	5	2,1	1, to 25
4	4	2,1	1 to
3			

ii) **Find the tight big Oh for the Best-case and Worst-case scenario.** Explain in one line how you drive it.

Best Case

In case n=0 then no loop ~~doe~~ runs.

≈ O(1)

Worst Case

In case n>0.

O(1) + O(N) + O(N) + O(N lg₂N) + O(N lg₂N) + O(N lg₂N) + O(N lg₂N) + O(N lg₂N) + O(N lg₂N) + O(N² lg₂N) + O(N² lg₂N)

≈ O(N² lg₂N)

i	j	k
5	5	1
	2	2
4	4	5
	2	1
3	3	1