

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

Question 1 [15 marks]

Select the correct answer among the multiple answers given in each part. Write the correct option (e.g. a or b etc.) in the table given below. Only the answer written in the given table below will be considered and there will be no marks for cutting.

option names are same in Q 3, 4

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
d	a	a	b	c	b	b	b	c	c	b	c	b	c	a

1. The given keys are inserted into an AVL tree: 1, 2, 3, 8, 6. How many rotations are performed?

- (a) 1 single rotation
- (b) 1 double rotation
- (c) 2 single rotations
- ☒ (d) 1 single rotation and 1 double rotation

RR
LR

2. In the following pseudo code, the parameter *start* is the root of a binary tree. What will the following code do to the tree?

```
pseudo_code(start)
  if (start==null)
    return 0;
  else
    return pseudo_code(start->right) + 1 + pseudo_code(start->left);
```

- ☒ (a) count the number internal nodes
- (b) count the number of leaf nodes
- (c) count the number of nodes who has both right and left child
- (d) mistake in the pseudo code

3. Any recursive function consists of two parts: a Base part that is non-recursive, where recursion stops and a recursive part.

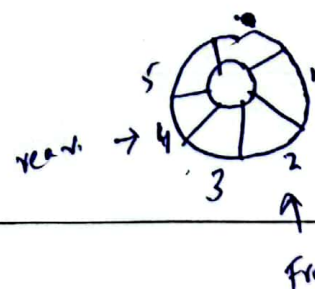
- ☒ (a) Base, Recursive
- (b) Base, Redundant
- (b) Recursive, Base
- (d) View, Base

4. Let a circular queue with a maximum size of six elements and the following front and rear positions
front = 2, rear = 4

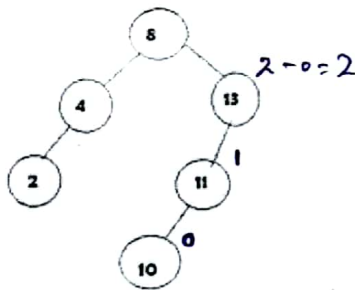
What will happen after an ADD another operation takes place?

6 elements

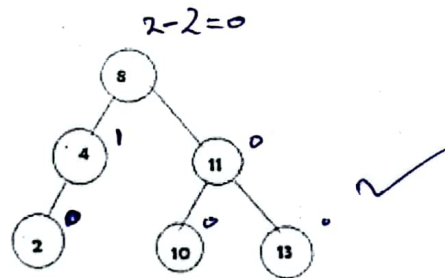
- a) front = 2 rear = 5
- ☒ (b) front = 3 rear = 4
- (c) front = 3 rear = 5
- (d) front = 2 rear = 4



5. If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue?
 a) $\text{rear} = (\text{rear} \% 1) + \text{MAX_SIZE}$
 b) $\text{rear} = \text{rear} \% (\text{MAX_SIZE} + 1)$
 c) $\text{rear} = (\text{rear} + 1) \% \text{MAX_SIZE}$
 d) $\text{rear} = \text{rear} + (1 \% \text{MAX_SIZE})$
6. The height of a BST is given as h. Consider the height of the tree as the no. of edges in the longest path from root to the leaf. The maximum no. of nodes possible in the tree is?
 a) $2^{h-1} - 1$
 b) $2^{h+1} - 1$
 c) $2^h + 1$
 d) $2^{h-1} + 1$
7. Suppose a binary tree is constructed with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be?
 a) $(n+1)/2$
 b) $(n-1)/2$
 c) $n/2 - 1$
 d) $(n+1)/2 - 1$
8. Which of the following statement about binary tree is CORRECT?
 a) Every binary tree is either complete or full
 b) Every complete binary tree is also a full binary tree
 c) Every full binary tree is also a complete binary tree
 d) A binary tree cannot be both complete and full
9. Which of the below diagram is following AVL tree property?



i



ii.

- a) only I
 b) both i and ii
 c) only ii
 d) i is not a binary search tree

10. Why do we need a binary tree which is height balanced?
 a) to avoid formation of skew trees
 b) to save memory
 c) to attain faster memory access
 d) to simplify storing

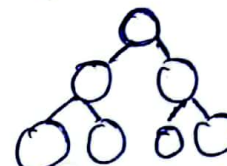
National University of Computer and Emerging Sciences
FAST School of Computing **Fall-2023** **Islamabad Campus**

11. Which type of traversal of binary search tree outputs the value in sorted order? ↑

- a) Pre-order *Root L R* c) Post-order *L R Root*
☒ b) In-order *L Root R* d) None

12. If every non-leaf node in a binary tree has nonempty left and right subtrees, the tree is termed a

- a) Full binary tree ☒ c) Complete binary tree
b) Binary tree d) Almost complete binary tree.



13. What is an AVL tree?

- a) a tree which is balanced and is a height balanced tree
☒ b) a tree which is unbalanced and is a height balanced tree
c) a tree with three children
d) a tree with at most 3 children

14. In a max-heap, element with the greatest key is always in the which node?

- a) Leaf node
b) First node of left sub tree
☒ c) root node
d) First node of right sub tree

15. Consider the pseudo code:

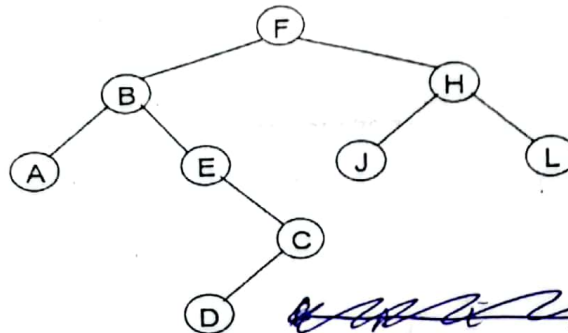
```
int avl (binarysearchtree root):
    if (not root)
        return 0
    left_tree_height = avl(left_of_root)
    if (left_tree_height == -1)
        return left_tree_height
    right_tree_height = avl(right_of_root)
    if (right_tree_height == -1)
        return right_tree_height
```

Does the above code check if a binary search tree is an AVL tree?

- ☒ a) yes
b) no

Question 2 [7 marks]

- a. Consider the binary tree shown below. For each of the traversals listed, give the order in which the nodes are visited. [4 marks]



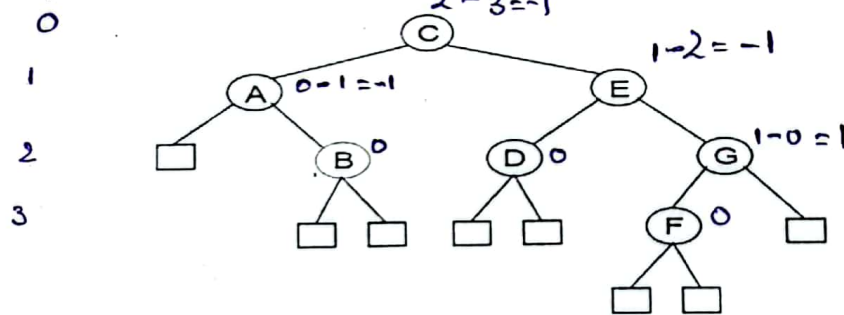
Postorder: D C A E B H F J L

Preorder	Root	L	R	C	D	B	A	E	F	H	J	L
Inorder												
Postorder												
Breadth-first (level order)												

Inorder

D C A B E F H J L

- b. For each node shown in the binary tree below, show its depth, height, and AVL balance factor. Write your answers in the following table. [3 marks]



Node

NULL

Not considering ☐ boxes.

Node	Depth	Height	Balance Factor
A	1	1	-1
B	2	0	0
C	0	3	-1
D	2	0	0
E	1	2	-1
F	3	0	0
G	2	1	1

National University of Computer and Emerging Sciences
FAST School of Computing **Fall-2023** **Islamabad Campus**

Question 3

Write a C++ function to find middle element of circular linked list in a single pass. You can traverse through the entire list only once. You are not allowed to use any global or static variables. [10 marks]

For example:

Input: 3 → 4 → 5 → 6 → 10 → 9 → 2 → 3 (circular)
 Output: Middle element: 6

Input: 5 → 2 → 8 → 1 → 3 → 6 → 4 → 7 → 5 (circular)
 Output: Middle element: 1

```
#include <iostream>
using namespace std;
class Node
{
public:
    int data;
    Node* next;
    Node* prev;
    Node(int d)
    {
        data = d;
        next = nullptr;
        prev = nullptr;
    }
};

class Circular
{
private:
    int count;
    Node* head;
    Node* tail;
public:
    Circular()
    {
        head = nullptr;
        tail = nullptr;
        count = 0;
    }
};
```

```
int main ()
{
    Circular c;
    c.findmiddle ();
}
```

Function is correct

```
① void findmiddle (int v, int s, int e)
{
    Node* curr = head;
    while (curr->next != head)
    {
        curr = curr->next;
        count++;
    }

    int start = head->data;
    int end = count;
    int mid;
    mid = (start + end) / 2;
    cout << mid;
}
```

OR

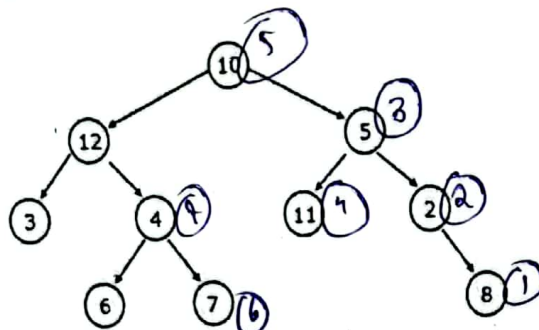
```
② void findmiddle (int v, int s, int e)
{
    int mid;
    mid = (start + end) / 2;
}
```

Query:- Implementation is all correct just we can not use

Function is correct
 Do not deserve 0

Question 4

In this question, you are required to write code for a fast-order traversal. For the given Tree below



R Root 2

when fast-order traversal is applied, it should print the nodes in following order:

8, 2, 5, 11, 10, 7, 4, 6, 12, 3

[10 marks]

```
#include <iostream>
using namespace std;
class Node
{ public:
    Node* right;
    Node* left;
    int data;
    Node (int a)
    { data = a;
      right = nullptr;
      left = nullptr;
    }
};
```

```
class fast_order
{
    Node* root;
    public:
    fast_order(Node* r)
    { root = r; }
```

```
void fast_traversal(Node* root)
{
    if (root == nullptr)
    { return; }
    fast_traversal(root->right);
    cout << root->data;
    fast_traversal(root->left);
}

int main ()
{
    fast_order f;
    fast_order f;
```


National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

Query: I have started from page 5.

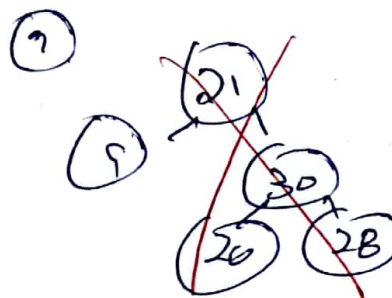
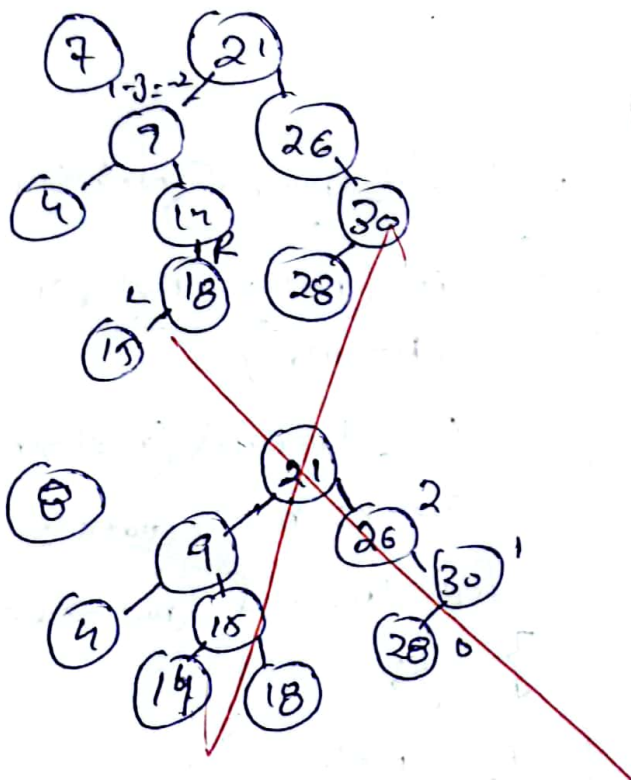
Question 5

If we construct the AVL Tree for the following Sequence

21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 3, 7

[5+3 marks]

- a) What will be the final AVL tree? You can use rough space to insert elements one by one and only make final tree here.



- b) Write number of rotations (RR, LL, RL, LR) performed to build the above tree

RR + LL	6
LR + RL	3

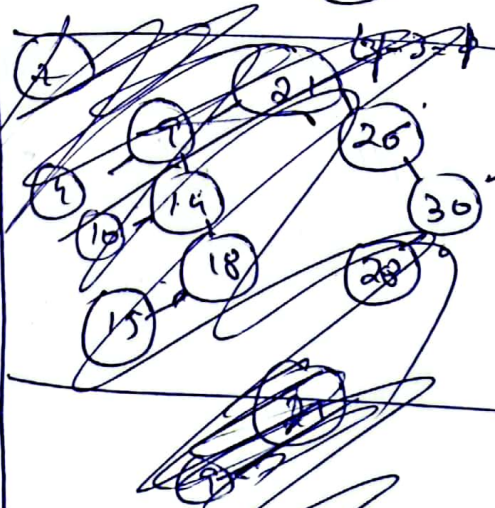
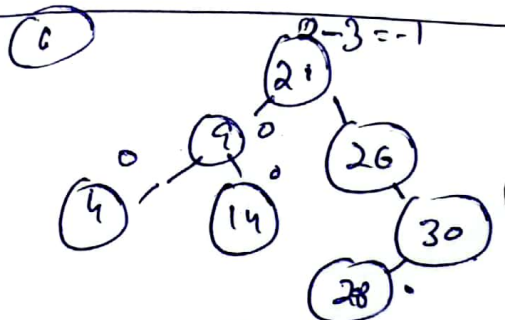
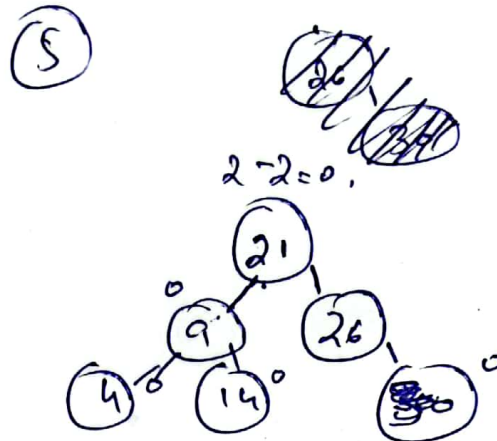
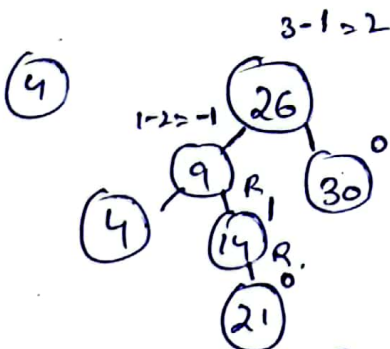
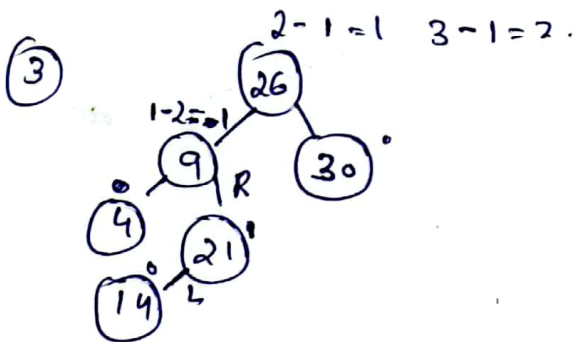
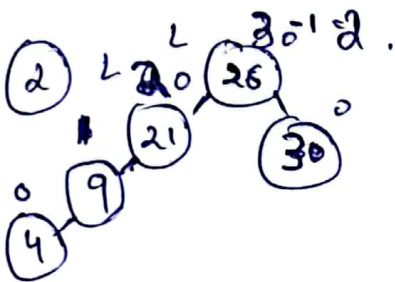
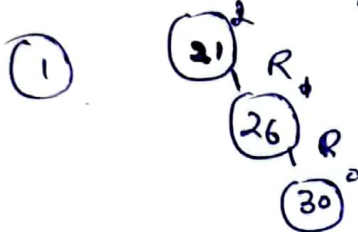
Start from here.

RR, LL, RL, RR,

0234

Question 5:-

~~21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 3, 7.~~
21, 26, 30, 9, 4, 14, 28, 18, 15, 10, 2, 3, 7.



Continued on
Page 4