Theory Final Exam

Marks

---

1. Write the time complexity of the following code segments with proper explanation. **10**

```
void fun(int l,int r)
{
	int mid = (l+r)/2;
	for(int i = l ; i <= r ; i++)
	{
		cout<<i<<endl;
	}
	if(l<r){
		fun(l,mid);
		fun(mid+1,r);
	}
}

int main()
{
	int n;
	cin>>n;
	fun(0,n-1);
}
```

```
for(int i = 1 ; i <= n/2 ; i++)
{
	for(int j = 1 ; j <= n ; j = j + i)
	{
		cout<<i<<" "<<j<<endl;
	}
}
```

2. Suppose you are implementing a linked-list where you want to maintain a floating point number and a character in each node. Each node will contain a next pointer and also a next_to_next pointer that will keep track of the node that is next to the next node. What will the node class look like? **10**

```
class Node{
	// write your variables
};
```

3. Write the main difference between linear and non-linear data structures. Compare between Stack, Queue and Deque. Are stack, queue, deque linear or non-linear data structure? What about a tree? **10**

4. Between singly linked list and doubly linked list which is better for implementing Stack and Queue? What about Deque? **10**

5. Convert the infix expression to postfix expression using a stack. You need to show all the steps. **10**

a*b+c*d+e

6. Compare the memory usage of Array, Singly Linked-list and Doubly Linked-list with necessary explanation. **10**

7. Suppose you are implementing a stack in a scenario where numbers are added in sorted order so that the stack is always sorted. Sometimes you need to quickly search if a value exists in the stack or not. Array or Linked-list which implementation for stack will you prefer in this scenario? Give necessary explanations. **10**

8. Suppose you are maintaining a head and tail for a singly linked-list. What will be time complexity of **10**
    a. Inserting a value at the beginning
    b. Inserting a value at the end
    c. Deleting a value at the beginning
    d. Deleting a value at the end
    e. Inserting a value at the mid point
    f. Deleting a value at the mid point

9. Consider the following binary tree in **Fig 1** (node 20 is the root) and answer the given questions. **10**

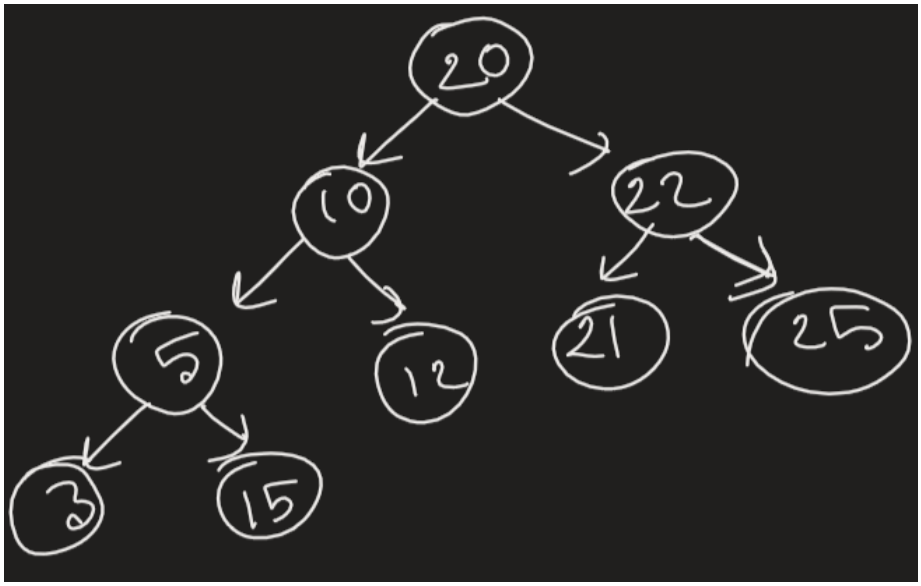

Fig: 1

a. Is the tree a Perfect binary tree? Why or why not?
b. Is the tree a Complete binary tree? Why or why not?
c. Is the tree a Binary search tree? Why or why not?
d. Write down the BFS, inorder, preorder and postorder traversal of the tree.

10. Write the steps to insert **70** in the following binary search tree in **Fig 2** (node 50 is the root). **10**

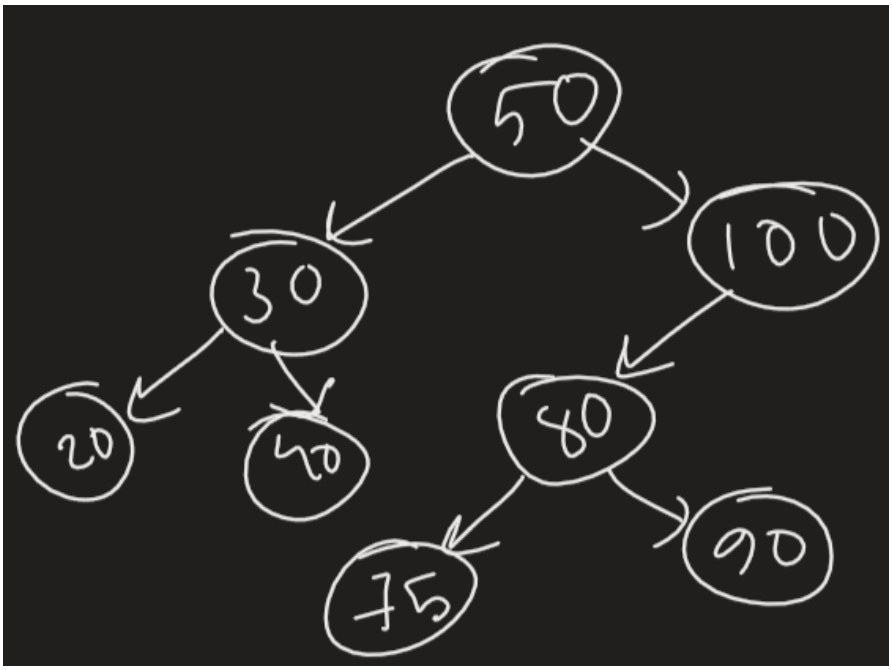

Fig: 2