**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;  }  } |

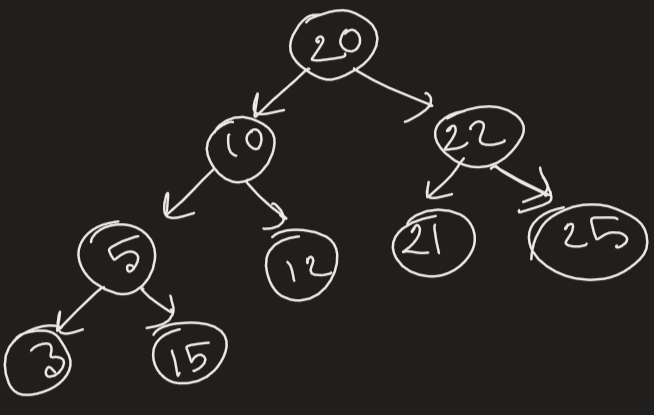
1. Suppose you are implementing a linked-list where you want to maintain a floating point number and a character in each node. Each node will contain a next pointer and also a next\_to\_next pointer that will keep track of the node that is next to the next node. What will the node class look like? **10**

class Node{

// write your variables

};

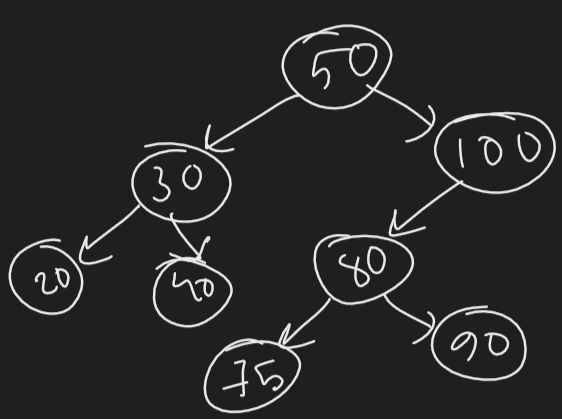
1. Write the main difference between linear and non-linear data structures. Compare between Stack, Queue and Deque. Are stack, queue, deque linear or non-linear data structure? What about a tree? **10**
2. Between singly linked list and doubly linked list which is better for implementing Stack and Queue? What about Deque? **10**
3. Convert the infix expression to postfix expression using a stack. You need to show all the steps. **10**  
    **a\*b+c\*d+e**
4. Compare the memory usage of Array, Singly Linked-list and Doubly Linked-list with necessary explanation. **10**
5. 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**
6. Suppose you are maintaining a head and tail for a singly linked-list. What will be time complexity of **10**
   1. Inserting a value at the beginning
   2. Inserting a value at the end
   3. Deleting a value at the beginning
   4. Deleting a value at the end
   5. Inserting a value at the mid point
   6. Deleting a value at the mid point
7. Consider the following binary tree in **Fig 1** (node 20 is the root) and answer the given questions. **10**



**Fig: 1**

1. Is the tree a Perfect binary tree? Why or why not?
2. Is the tree a Complete binary tree? Why or why not?
3. Is the tree a Binary search tree? Why or why not?
4. 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**