

Answering Question 1 is mandatory, Question 2 or 3, Question 4 or 5 (Total 3 answers)

**Department of Computer Science and Engineering**  
**MIDTERM EXAMINATION, Summer' 19**  
**CSE 221: Algorithms**

**Total Marks: 30    Time Allowed: 50 minutes**

**Student ID :**

**Section :**

**Name:**

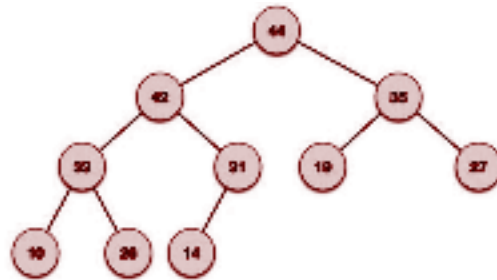
**Question 1 [CO1]: Visualize** the following graphs according to the description below and represent them using the mentioned data structure. [10]

(a) Draw a directed weighted graph with 6 vertices, 9 edges and it contains at least 2 cycles. Assign random weights to the edges.	(b) Draw a connected undirected graph of 6 vertices with a minimal number of edges.	(c) Draw a strongly connected graph with 6 vertices.
Adj. Matrix: ?	Adj. List: ?	Adj. List: ?

**Question 2 [CO2]**

[4+3+3 = 10]

a) The tree on the right is a max heap. Write the pseudocode and **Show** how to insert a new node containing the key “46” while maintaining the heap property. Show all the steps clearly.



b) **Show** how the mergesort algorithm is used to sort the following array and **show** the time complexity of merge sort algorithm is  $O(n \lg n)$ .

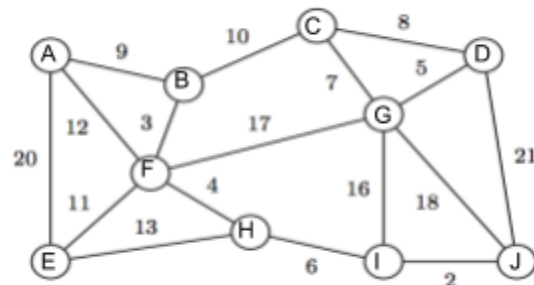
31	12	23	2	7	9	51	22	15
----	----	----	---	---	---	----	----	----

**Question 3 [CO2]**

[8 + 2 = 10]

(i) **Show** simulation of a suitable algorithm to determine MST for the following graph. Find out the cost of the MST.

(ii) **Identify** another algorithm you can apply instead of the one you applied above? What is the running time of both algorithm? In which situations will one algorithm performs better than the other?



Answering Question 1 is mandatory, Question 2 or 3, Question 4 or 5 (Total 3 answers)

**Question 4 [CO5]**

[10]

An array of numbers is bitonic if it consists of a strictly increasing sequence followed by a strictly decreasing sequence e.g.

1, 2, 4, 5, 3, 2

```
public static int AlgoX(int a[], int low, int high)
{
    if(low == high) return high;
    int mid = low + (high - low)/2;
    if(a[mid] < a[mid+1])
        return AlgoX(a, mid, high);
    else if(a[mid] > a[mid+1])
        return AlgoX(a, low, mid);
    else return mid;
}
```

Algorithm Explanation:

The **maximum** can be found in Log N comparisons using binary search. Each step compares two adjacent numbers A[i] and A[i+1]. If they are equal, they are both maximum. If A[i] is smaller, we search to the right of i, if A[i] is bigger, we search to the left of i+1.

- Design** the code to solve this using the partially completed code on the left.
- Write the recurrence equation and find the time complexity

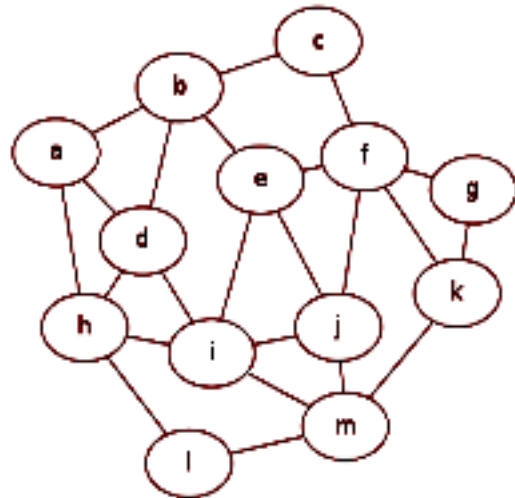
**Question 5 [CO5]**

[8+2 =10]

a) Piku is very interested in graphs and algorithms. He recently learned using traversal algorithms, using which he can determine if there is a path between two nodes in a graph or not.

**Modify** a suitable algorithm for Piku to find if there is a path between two nodes or not (Algorithm will return true if there is a path and false otherwise). Show the simulation step by step to find out the path from node 'b' to node 'm'.

b) **Explain** DAG with an example.



Answering Question 1 is mandatory, Question 2 or 3, Question 4 or 5 (Total 3 answers)

Alternate Questions:

### Question 5

a) Piku is very interested in graphs and algorithms. He is also a very well organized person. Everyday he makes a graph of the tasks which need to be done and enumerates them from m to z. However, some tasks must be done before others. Help Piku find a correct order for each of the tasks he needs to complete.

**Show** the simulation step by step to find out the order of tasks.

b) Of all the algorithms you've learned so far, which algorithm is appropriate for finding the shortest path for an unweighted graph? **Explain** DAG with an example.

