This question paper contains 4 printed pages.

Your Roll No. ....

F-6

Sl. No. of Ques. Paper	**: 6367**
Unique Paper Code	**: 2341401**
Name of Paper	**: Design and Analysis of Algorithms**
Name of Course	**: B.Tech. in Computer Science**
Semester	**: IV**
Duration :	**: 3 hours**
Maximum Marks	**: 75**

(Write your Roll No. on the top immediately on receipt of this question paper.)

Q. No. 1 (35 marks) is compulsory. Attempt four questions from Q. No. 2 to Q. No. 7.

1.(a) Given a graph G= (V, E). Write an algorithm to determine if the graph is acyclic. (4)

(b) Let P be a shortest path from some vertex s to some other vertex t in a graph. If the weight of each edge in the graph is increased by one, will P still be a shortest path from s to t? Support your answer with appropriate arguments. (3)

(c) Assuming adjacency list representation of graphs, discuss the time complexity of performing BFS. (3)

(d) Consider the following recurrence relation: (4)

$f(n) = 2, \quad \text{if } n <= 1$
$f(n) = f(n-2) + n, \quad \text{if } n > 1.$

Write a dynamic programming algorithm to compute f(n) above.

(e) Do greedy algorithms always give optimal soutions? If not, give an example where the greedy algorithm does not always give the optimal solution. (3)

(f) Describe an algorithm that given n integers in the range 0 to k preprocesses its input and answers any query about how many of the n integers fall into range [a, b] in O(1) time. Preprocessing time should not be more than Θ(n+k). (4)

(g) Prove that red-black trees are balanced, i.e., if a red-black tree contains n nodes, then its height is O(log n). (5)

(h) Insertion sort can be expressed as a recursive procedure as follows. Given A[1..n], we recursively sort A[1..n-1] and then insert element A[n] into the sorted array A[1..n-1]. Write a recurrence relation for the running time of this recursive version of insertion sort and give time complexity of the algorithm by solving the recurrence. (4)

P.T.O

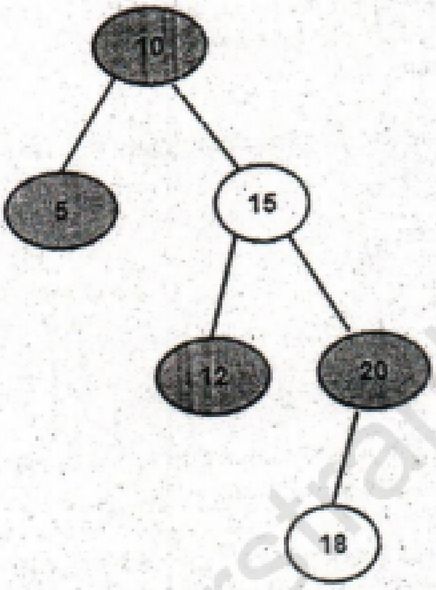
(i) Analyze time complexity for the given algorithm (written in pseudocode) (3)

```
x = 0;
i = 1;
while( i < n )
{
  i = 2 * i;
  x++;
}
```

(j) State whether the sequence <20, 15, 18, 7, 9, 5, 12, 3, 6, 2> a max-heap or not? (2)

2.(a) Can we use Heapsort as the auxiliary sorting routine in radix sort? (2)

(b) Consider the following Red-Black Tree: (6)



Insert 7, 17 in the given red-black tree. From the resultant tree, obtained after inserting the two new nodes, delete 17 and then 7.

(c) What is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number of nodes? (2)

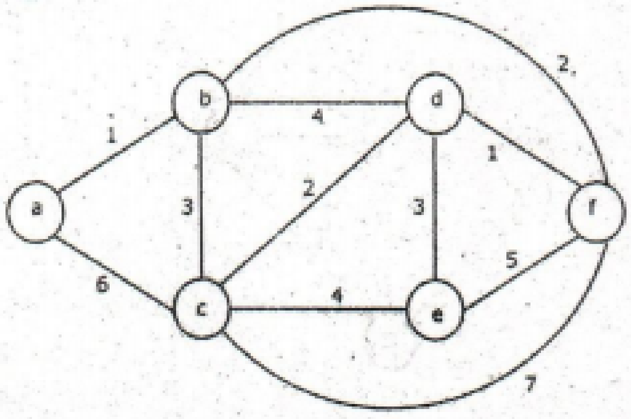
3.(a) Two key ingredients that an optimisation problem must have in order for dynamic programming to apply are optimal substructure and overlapping subproblems. Explain these two properties. Show that rod-cutting problem has both these properties. (3+2)

(b) Determine an LCS of <1,0,0,1,0,1,0,1> and <0,1,0,1,1,0,1,1,0>. (5)

4.(a) Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered from 1 to 7 as in a binary heap. (3)

6367

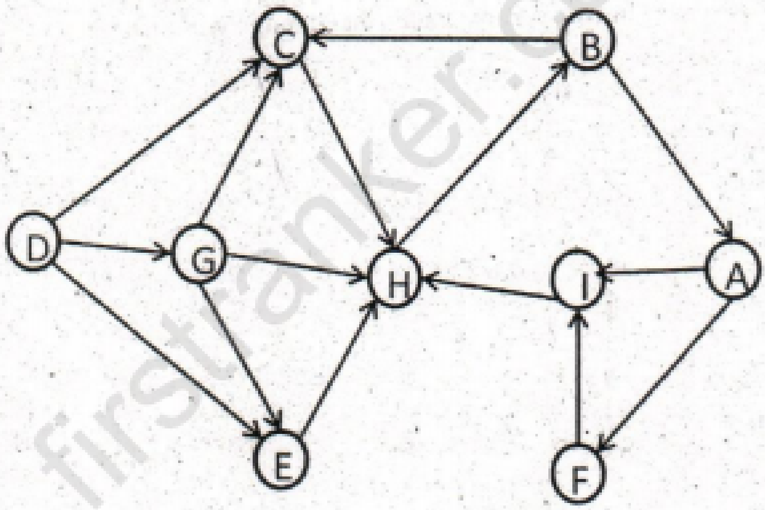
(b) Find a minimum spanning tree for the weighted graph shown below using Prim's MST algorithm: (4)



(c) Consider the following graph. (3)

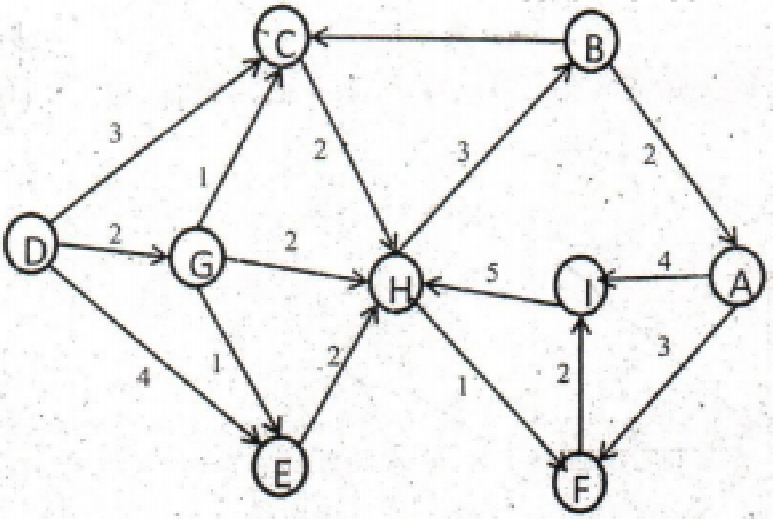
i) Draw the resultant depth-first tree obtained on running DFS on the graph. Start your search with vertex B.

i) Show the discovery and finishing times for each vertex, and show the classification of each edge



5.(a) Given an adjacency list representation of a directed graph. Give an efficient algorithm to compute the transpose. Analyse the running time of your algorithm. Transpose of a graph G= (V,E) is the graph $G^T = (V, E^T)$, (5)

(b) Find a shortest path from vertex D to vertex A in the following graph by carefully following the steps of Dijkstra's algorithm. (5)



6.(a) Show that the second smallest of n elements can be found with $n + \lceil \lg n \rceil - 2$ comparisons in the worst case. (4)

(b) What do you understand by a stable algorithm? Is the following code of count sort stable? Explain why/why not. In case it is unstable what change will make it stable? (3)

```
count sort(A,B,k)
   for i=0 to k
      C[i]=0
   for j=1 to A.length
      C[A[j]]=C[A[j]]+1
   for i=1 to k
      C[i]=C[i]+C[i-1]
   for j=1 to A.length
     B[C[A[j]]] = A[j]
     C[A[j]] = C[A[j]] – 1
```

(c) Consider sorting n numbers stored in array A using Selection Sort. Why does it need to run for only the first n - 1 elements, rather than for all n elements? Give the best-case and worst-case running times of selection sort in Θ-notation. (3)

7.(a) Let A[1 . . n] be an array of n distinct numbers. If i < j and A[i] > A[j], then the pair (i, j) is called an ***inversion*** of A. (2+3)

i) List the five inversions of the array <2, 3, 8, 6, 1>.

ii) What array with elements {1, 2, …, n} has the most inversions? How many does it have?

(b) Give a pattern beginning with A and using only letters from the set {A,B} that would have the following failure indexes (for KMP flowchart construction) (5)
Fail= < 0,1,1,2,3,4,2,2>

400