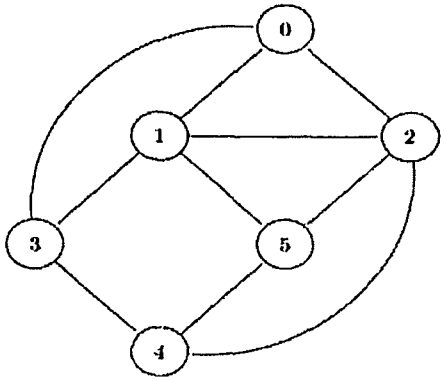
(c) Give the DFS traversal sequence from vertex 0 for the following graph : (4)



[This question paper contains 12 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 13359 **K**

Unique Paper Code : 2344000030

Name of the Paper : Design And Analysis of Algorithms

Name of the Course : **Generic Elective GE**

Semester : VII

Duration : 3 Hours Maximum Marks : 90

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. The Paper has **two** sections.
3. All questions in **'Section A'** are compulsory.
4. Attempt any **four** questions from **'Section B'**.

SECTION A

1. (a) Given keys {26, 30, 42, 35, 50} and table size = 7, insert the keys using hash function h(k) = k mod 7 and resolve collisions using linear probing. (3)

(b) Find the total number of comparisons made by the insertion sort algorithm for the following array A:

A = [8, 4, 2, 9, 5] (3)

(c) Can a greedy algorithm always give the optimal solution for all optimization problems? Justify your answer with an example. (3)

(d) (i) What is the time complexity of the Heap Sort algorithm?

(ii) How many comparisons does the Heap Sort algorithm make in order to sort an already sorted array? (3)


```
func(n - 1, source, dest, aux);
cout << "Move disk" << n << " from " << source
<< " to " << dest << endl;
func(n - 1, aux, source, dest);
}
```

(ii)
```
void func(int n){
if(n <= 1) return;
func(n/3);
func(2n/3);
}
```

(b) Given 4 intervals (1,4), (3,5), (0,6) and (5,7) each representing start and end times of an activity requiring the same resource. Find an optimal solution of non-overlapping intervals using the following strategies

(i) earliest start time first.

(ii) shortest duration first.

(iii) earliest finish time first. (6)

P.T.O.

(b) (i) Why does the greedy algorithm fail for weighted interval scheduling?

(ii) Give an example where selecting the interval with the earliest finishing time does not yield the optimal total value. (5)

(c) Will Dijkstra's algorithm still give the shortest path between two vertices if the edge weights are allowed to be negative? If yes, justify your answer with an argument. If not, give an example. (3)

7. (a) Give a recurrence relation for the following functions : (5)

(i)
```
void func(int n, char source, char aux, char dest) {
if (n == 1){
cout << "Move disk 1 from " << source << " to "
<< dest
<< endl;
return;
}
```


(e) Can a graph with a cycle have a Topological ordering? Justify your answer with a short example. (3)

(f) Which algorithm design approach gives optimal solution for the following : (3)

(i) Binary Search

(ii) Interval Scheduling Problem

(iii) Subset Sum Problem

(g) Sort the following list of integers using Radix Sort

A = [329, 457, 657, 839, 436]

Show the state of the array after each digit-level sorting pass. (3)

(h) (i) What is an adjacency matrix?

(ii) State one advantage and disadvantage of representing a graph using adjacency matrix. (3)

P.T.O.

(i) Define strongly connected graph in the context of directed graphs. Draw a strongly connected graph with five nodes. (3)

(j) (i) What is the selection problem?

(ii) What is the least number of comparisons to find the minimum and maximum element in an array. (3)

SECTION B

2. (a) Explain how Strassen's algorithm is better than traditional matrix multiplication. (3)

(b) Match the following sorting algorithms (Column A) with their correct properties (Column B).

Column A	Column B
A. Count Sort	I. Stable but not in-place
B. Insertion Sort	II. Stable and in-place
C. Merge Sort	III. Not stable but in-place
D. Quick Sort	IV. Uses divide and Conquer, stable, and requires extra memory (4)

(c) Consider the following recursive algorithm to find an optimal 5 schedule for the weighted interval scheduling problem :

Compute_opt(j)

If j = 0, then

return 0

Else

return max(v_j + Compute_opt(p(j)), Compute_opt(j-1))

(i) Explain why this algorithm takes exponential time to run in the worst case.

(ii) What changes should be made to the above algorithm to make it run in polynomial time?

6. (a) Apply Merge sort on the following array :

[12, 4, 8, 3, 9, 7,1]

Show the recursion tree and all intermediate steps. (7)

(c) (i) Explain the Divide and Conquer technique.

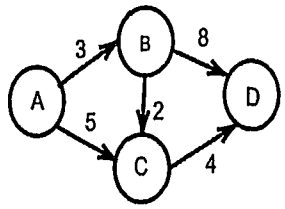
(ii) Is Quick sort based on the Divide and Conquer strategy?

(iii) What is the running time of Quick sort when all elements of the array have the same value? (5)

5. (a) (i) Define optimal substructure and overlapping subproblems.

(ii) Explain how Dynamic Programming uses these properties with reference to the Fibonacci number computation. (5)

(b) Show the step-by-step execution of Dijkstra's algorithm on the 5 following graph, taking vertex A as the source vertex : (5)



(c) (i) Write the algorithm for Binary Search.

(ii) Apply Binary Search algorithm to find the key value 31 from the given array.

A = [10, 14, 19, 26, 27, 31, 33, 35, 42, 44].

Show the value of low, high, and mid at each step.

(iii) How many comparisons are made to search for the key "42" using binary search on array A.

3. (a) Solve the following 0/1 Knapsack problem using Dynamic Programming : (5)

Items	Values (₹)	Weights (kg)
1	(6)0	10
2	100	20
3	120	(3)0

Capacity (W): 50 kg

Show the DP table and find the maximum profit.

P.T.O.

(b) (i) What is chaining in the context of hashing?

(ii) Consider a hash table of size 9 with the hash function

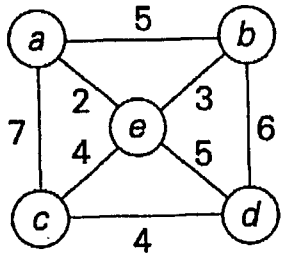
$$h(k) = k \bmod 9$$

Insert the following keys using the chaining method:

19, 26, 13, 46, and 37.

Show the final hash table representation. (5)

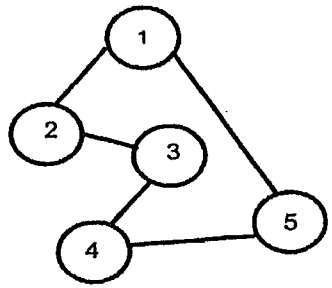
(c) Show the step-by-step execution of Prim's algorithm on the following graph starting with vertex a : (5)



4. (a) (i) Define a bipartite graph. (5)

(ii) How does the two-coloring principle help in identifying bipartite graphs?

(iii) Given the graph :



Determine whether it is bipartite.

(b) Let G = (V, E) be a connected, weighted graph, $v \in V$ be any vertex, and e be an edge with minimum weight among all the edges that are incident on v. Prove that there exists an MST that includes the edge e. (5)

P.T.O.