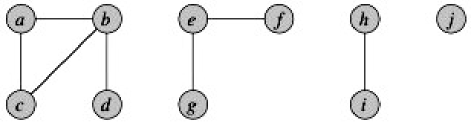
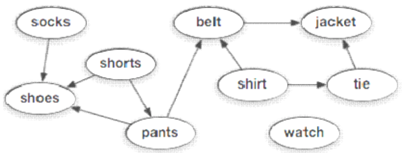
(b) Find out the number of components for the following graph using union-find (union by rank),



Explain the subroutines used in detail with their time complexities.

(4 + 4) + 4 = 12

7. (a) Topologically sort (draw the DAG) the following graph with its vertices arranged from left to right in order of decreasing finishing time.



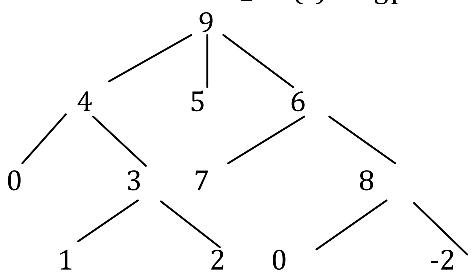
(b) Prove that shortest paths are made of shortest paths.

9 + 3 = 12

Group – E

8. (a) Define flow in a flow network. What is an s-t cut of a flow network? Prove that flow of any s-t cut is equal to the flow of the network.

(b) Please draw the tree after the call of FIND_SET(3) using path compression heuristic.



Write one application of disjoint set data structure.

(2 + 2 + 4) + (3 + 1) = 12

9. (a) Explain with a small diagram how you can show the easiness or hardness of a new problem using the easiness or hardness result of a known problem.

(b) Show that the Clique Decision Problem is NP-hard by using the result that the 3-CNF-SAT problem is computationally hard.

(c) Define the following terms in the context of optimization problems —
(i) Approximation scheme
(ii) Polynomial-time approximation scheme
(iii) Fully polynomial-time approximation scheme (FPTAS).

3 + 5 + (2 + 1 + 1) = 12

DESIGN & ANALYSIS OF ALGORITHMS
(CSEN 2201)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) Time complexity to identify Minimum Spanning Tree of a graph G(V, E) using binary heap is
(a) Θ(V+E) (b) O(V.E)
(c) O(E log V) (d) none of these.

(ii) The worst case time complexity of the following recurrence will be, $T(n) = 3T(n/2) + 3(n/4)+ 1$
(a) $O(n^2)$ (b) $O(n^2 \log n)$
(c) $O(n \log n)$ (d) $O(n^{1.58})$.

(iii) Consider the following code snippet:

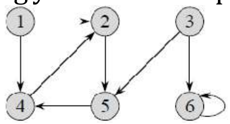
```
int fun(int n)
{
 int count = 0;
 for (int i = 0; i < n; i++)
   for (int j = i; j > 0; j--)
     count = count + 1;
 return count;
}
```

What is the time complexity of fun() in worst case?
(a) θ (n) (b) θ (n^2)
(c) θ (n*logn) (d) θ (nlognlogn)

(iv) Let *S* be a sorted array of n integers. Let t(n) denote the best case time taken by an algorithm to determine if there are two elements with sum less than 1000 in *S*. which of the following statements is true?
(a) t(n) is O(1) (b) $n < t(n) < n\log_2 n$
(c) $n\log_2 n < t(n) < \binom{n}{2}$ (d) $t(n) = \binom{n}{2}$

(v) After running Bellman-Ford algorithm, if, for any edge e(u,v) with weight w(u,v), we see that v.d ≤ u.d + w(u,v) is satisfied, we can conclude that,
(a) there is at least one –ve weight edge
(b) there may be a -ve weight edge, but there is no -ve weight cycle
(c) there is no -ve weight edge as well as -ve weight cycle
(d) there is at least one –ve weight.

(vi) Count the number of strongly connected components from the following diagram,



(a) 2 (b) 3 (c) 4 (d) none of these.

(vii) Floyd-Warshall algorithm was applied on a directed graph G, where G.V = {1, 2, 3, 4, 5}. As a part of the final result, the predecessor matrix π(5) was generated. What will be the shortest path from source node 3 to target node 5?

$$\begin{pmatrix} NIL & 3 & 4 & 5 & 1 \\ 4 & NIL & 4 & 2 & 1 \\ 4 & 3 & NIL & 2 & 1 \\ 4 & 3 & 4 & NIL & 1 \\ 4 & 3 & 4 & 5 & NIL \end{pmatrix}$$

(a) 3-5 (b) 3-1-5
(c) 3-2-1-5 (d) 3-2-4-1-5.

(viii) The minimum number of comparisons required to sort a set of five distinct integers, if they are allowed to come in any arbitrary order, is
(a) 7 (b) 8 (c) 9 (d) 10.

(ix) The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of
(a) N (b) N^2 (c) NlogN (d) N(logN)^2.

(x) Space required to solve the LCS problems for two strings of size m and n, will be
(a) O(m+n) (b) O((m+n)logn) (c) $O(mn^2)$ (d) O(mn).

Group – B

2. (a) Solve the following recurrence by recurrence tree method and verify it with forward substitution method, $T(n) = 3T(n/4) + cn^2$

(b) Show that $\lceil 3n/2 \rceil - 2$ comparisons are necessary for finding the minimum and maximum of n numbers.

(c) Now show that $\lceil 3n/2 \rceil - 2$ comparisons are also sufficient for finding the minimum and maximum of n numbers.

4 + 6 + 2 = 12

3. (a) You are provided as input an even positive integer $n > 1$ and an unsorted sequence S of n integers. You would like to find *both* the

second order statistic and the nth order statistic of S. Show that they can be found out in $(3n/2 + \lceil \lg n \rceil - 3)$ number of pair-wise comparisons.
Hint: We can write $(3n/2 + \lceil \lg n \rceil - 3) = 3n/2 - 2 + \lceil \lg n \rceil - 1$

(b) Calculate the average-case time complexity for binary search.

7 + 5 = 12

Group – C

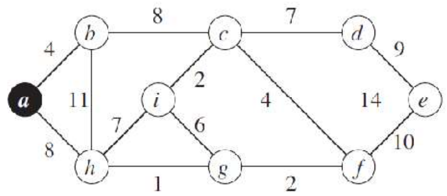
4. Describe how matrix-chain multiplication problem can be solved with dynamic programming. Explain the optimal substructure property and write down the pseudo-codes for calculating minimum number of scalar multiplications and finding optimal parenthesization.
Solve the problem for the given p = {35, 30, 5, 20, 15, 10}.

(1 + 4 + 3 + 4) = 12

5. (a) Please prove that longest common subsequence qualifies as a problem that can be solved efficiently using the dynamic programming technique.

(b) Solve the following problem using a greedy strategy. If you think, you cannot get optimal solution of the following problem using greedy strategy, please state the reason. If you think, otherwise, please prove that your strategy gives optimal solution.
Input: n sorted arrays of length L[1], L[2],...,L[n]
Problem: To merge the arrays pair wise as fast as possible. The problem is to determine which pair to merge every time.

(c) Show how Prim's algorithm works on the following graph by drawing a min-priority queue and (by repeatedly drawing) how it evolves with increase in "black" nodes in a step-by-step manner.



4 + 4 + 4 = 12

Group – D

6. (a) In a stack, two fundamental stack operations, each of which takes $O(1)$ time, are:
(i) PUSH(S, x) - pushes object x onto stack S and
(ii) POP(S) -pops the top of stack S and returns the popped object.
If we add the stack operation MULTIPOP(S, k), which removes the k top objects of stack S, or pops the entire stack if it contains less than k objects, find out the amortized cost of a stack operation.