Design & Analysis of Algorithms (PCC-CS-404) PYQs analysis

Based on: Dec 2016, 2017, 2023, May 2019, 2020, 2023, 2024

- Syllabus:
  - Module 1: Introduction
    - Characteristics of algorithms
    - Asymptotic analysis (best, average, worst-case behavior)
    - Performance measurements (time and space trade-offs)
    - Analysis of recursive algorithms (Substitution method, Recursion tree method, Master's theorem)
  - Module 2: Fundamental Algorithmic Strategies
    - Brute-Force
    - Greedy
    - Dynamic Programming
    - Branch-and-Bound
    - Backtracking
    - Problem examples: Bin Packing, Knapsack, Job Sequencing with Deadline, Optimal Binary Search Tree, N-Queen Problem, Hamiltonian Cycle, Traveling Salesman Problem (TSP)
    - Heuristics and their application domains
  - Module 3: Graph and Tree Traversal Algorithms
    - Depth First Search (DFS)
    - Breadth First Search (BFS)
    - Shortest Path Algorithms
    - Transitive Closure
    - Minimum Spanning Tree
    - Topological Sorting
    - Network Flow Algorithm

- **Module 4: Tractable and Intractable Problems**
  - Computability of algorithms
  - Computability classes (P, NP, NP-complete, NP-hard)
  - Cook's theorem
  - Standard NP-complete problems
  - Reduction techniques
- **Module 5: Advanced Topics**
  - Approximation algorithms
  - Randomized algorithms
  - Problems beyond NP (P-SPACE)

- **Repeated Questions** (identical wording, topic, approach):
  - **Recurrence Relation:** (T(n) = T(n^{1/2}) + c)
    - ***Years:*** 2016 (2 marks), 2019 (1.5 marks).
    - Text: Solve the recurrence relation (T(n) = T(n^{1/2}) + c).
    - ***Topic:*** Module 1 – Analysis of recursive algorithms.
    - *Marks:* 3.5 total.
  - **Job Sequencing with Deadline:** (n = 7, p = (3, 5, 20, 18, 1, 6, 30), d = (1, 3, 4, 3, 2, 1, 2)).
    - ***Years:*** 2016 (10 marks), 2017 (10 marks).
    - *Text:* Schedule jobs to optimize using Greedy method.
    - ***Topic:*** Module 2 – Job Sequencing with Deadline (Greedy).
    - *Marks:* 20 total.
  - **Knapsack Problem:**
    - ***2016:*** Fractional, Branch and Bound, (m = 60, w = (5, 10, 20, 30, 40), p = (30, 20, 100, 90, 160)), 10 marks.
    - ***2017:*** Greedy, (m = 30, w = (10, 15, 6, 9), p = (2, 5, 8, 1)), 10 marks.
    - ***2019:*** Greedy, (n = 7, m = 15, p = (10, 5, 15, 7, 6, 18, 3), w = (2, 3, 5, 7, 1, 4, 1)), 15 marks.
    - ***2020:*** 0/1 and Fractional, Greedy, (m = 30, w = (10, 15, 6, 9), p = (2, 5, 8, 1)), 15 marks.
    - ***2023 May:*** 0/1, Dynamic Programming, (m = 5, w = (2, 3, 4, 5), p = (3, 4, 5, 6)), 15 marks.
    - ***2023 Dec:*** Branch and Bound, (n = 4, p = (10, 10, 12, 18), w = (2, 4, 6, 9), m = 15), 7 marks.

    - *2024 May:* Branch and Bound, (m = 30, w = (18, 15, 8, 12), p = (36, 24, 8, 12)), 10 marks.
    - *Topic*: Module 2 – Knapsack (Greedy, Branch and Bound, Dynamic Programming).
    - *Marks*: 82 total.
  - Greedy vs. Dynamic Programming Comparison:
    - *Years:* 2016 (2 marks), 2017 (2 marks), 2023 May (1.5 marks), 2023 Dec (1.5 marks), 2024 May (1.5 marks, why Greedy not suggested for 0/1 Knapsack).
    - *Text:* Differentiate Greedy and Dynamic Programming methods or related limitations.
    - *Topic:* Module 2 – Greedy, Dynamic Programming.
    - *Marks:* 8.5 total.
  - P, NP, NP–Complete, NP–Hard:
    - *Years:* 2016 (2 marks, P vs. NP), 2017 (2 marks, NP–Complete vs. NP–Hard), 2019 (7.5 marks), 2020 (1.5 marks), 2023 May (7 marks), 2023 Dec (7 marks), 2024 May (1.5 marks, differentiate P, NP–Hard, NP–Complete).
    - *Text:* Discuss/define P, NP, NP–Complete, NP–Hard with examples.
    - *Topic:* Module 4 – Computability classes.
    - *Marks:* 28.5 total.
  - BFS vs. DFS Comparison:
    - *Years:* 2019 (1.5 marks), 2023 May (part of 15 marks), 2023 Dec (7 marks).
    - *Text:* Differentiate BFS and DFS graph traversal algorithms.
    - *Topic:* Module 3 – DFS, BFS.
    - *Marks:* ~13.5 total.
- Similar Questions (Same topic, minor deviations):
  - Knapsack Problem:
    - *2016:* Fractional, Branch and Bound, (m = 60, w = (5, 10, 20, 30, 40), p = (30, 20, 100, 90, 160)), 10 marks.
    - *2017:* Greedy, (m = 30, w = (10, 15, 6, 9), p = (2, 5, 8, 1)), 10 marks.
    - *2019:* Greedy, (n = 7, m = 15, p = (10, 5, 15, 7, 6, 18, 3), w = (2, 3, 5, 7, 1, 4, 1)), 15 marks.
    - *2020* 0/1 and Fractional, Greedy, (m = 30, w = (10, 15, 6, 9), p = (2, 5, 8, 1)), 15 marks.
    - *2023 May* 0/1, Dynamic Programming, (m = 5, w = (2, 3, 4, 5), p = (3, 4, 5, 6)), 15 marks.
    - *2023 Dec* Branch and Bound, (n = 4, p = (10, 10, 12, 18), w = (2, 4, 6, 9), m = 15), 7 marks.

- *2024 May:* Branch and Bound, (m = 30, w = (18, 15, 8, 12), p = (36, 24, 8, 12)), 10 marks.
- *Topic*: Module 2 – Knapsack (Greedy, Branch and Bound, Dynamic Programming).
- *Marks*: 82 total.

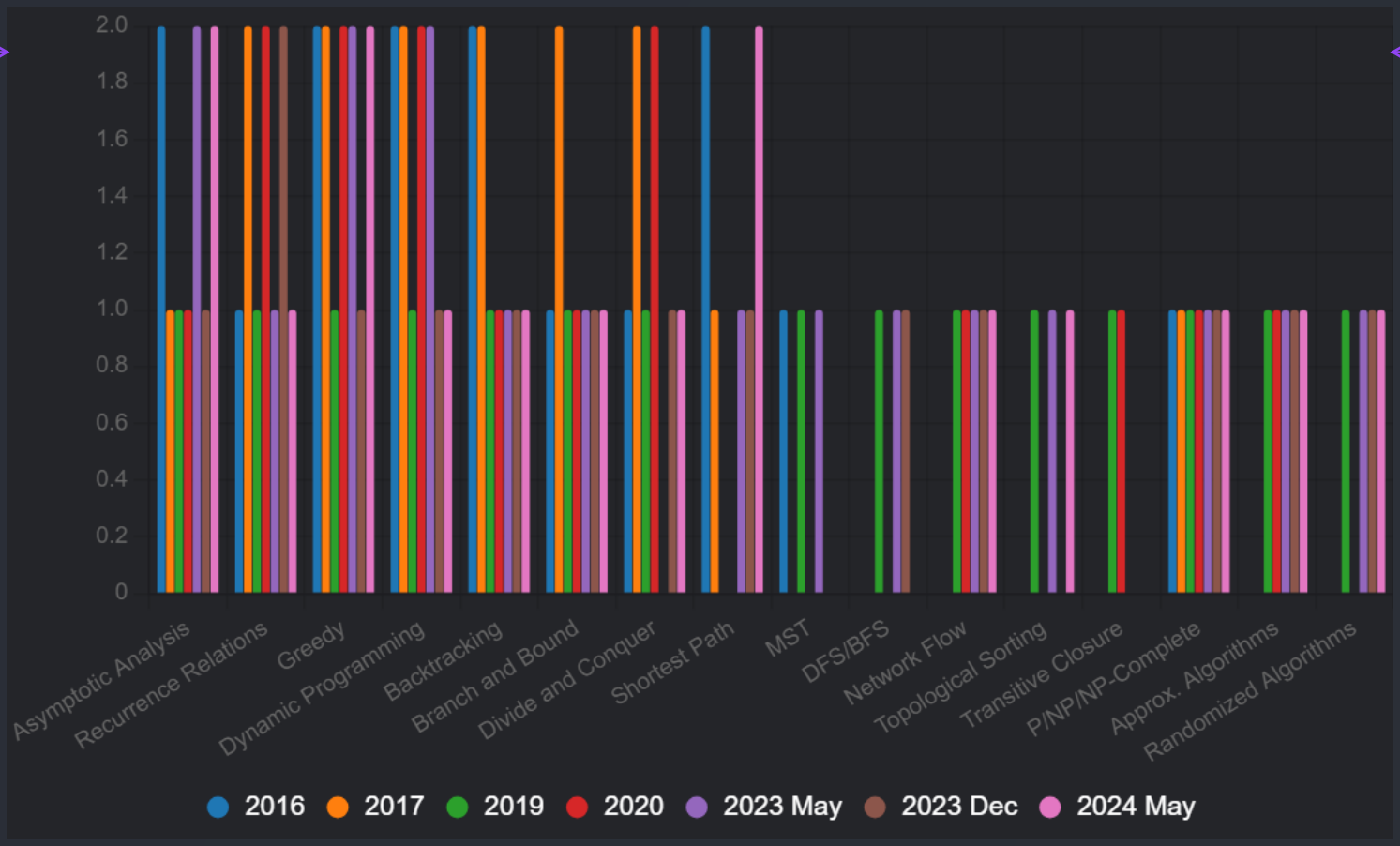
- **Traveling Salesman Problem (TSP):**
  - *2016:* Branch and Bound, 10 marks.
  - *2017:* Branch and Bound, 10 marks.
  - *2019:* Approximation algorithm, 15 marks.
  - *2020:* Branch and Bound, 15 marks.
  - *2023 May:* Branch and Bound, 15 marks; NP-Complete nature, 15 marks.
  - *2024 May:* Branch and Bound, 15 marks (optimal route for cities A, B, C, D).
  - Topic*: Module 2 – TSP; Module 4 – NP-Complete.
  - *Marks*: 80 total.
- **Shortest Path Algorithms:**
  - *2016:* Single Source vs. All-Pair time complexity (2 marks); All-Pair time complexity (2 marks).
  - *2017:* Single Source vs. All-Pair differentiation (2 marks).
  - *2019:* All-Pair Shortest Path algorithm (part of 15-mark question).
  - *2023 May:* Dijkstra's failure on negative weights (1.5 marks).
  - *2023 Dec:* Single Source Shortest Path algorithm (7.5 marks).
  - *2024 May:* Shortest path using Greedy (5 marks); Bellman-Ford negative weight cycles (1.5 marks).
  - *Topic*: Module 3 – Shortest Path Algorithms.
  - *Marks*: ~30 total.
- **N-Queen Problem:**
  - *2016:* Discuss Backtracking (2 marks).
  - *2017:* Implicit/explicit constraints (2 marks).
  - *2020:* Backtracking for 4x4 (15 marks).
  - *2023 May:* Brute-Force and Backtracking, time complexity comparison (15 marks).
  - *2023 Dec:* Backtracking algorithm, constraints (8 marks).
  - *2024 May:* Backtracking, differentiate with Dynamic Programming (7.5 marks).
  - *Topic*: Module 2 – N-Queen Problem (Backtracking).

- *Marks*: ~49.5 total.

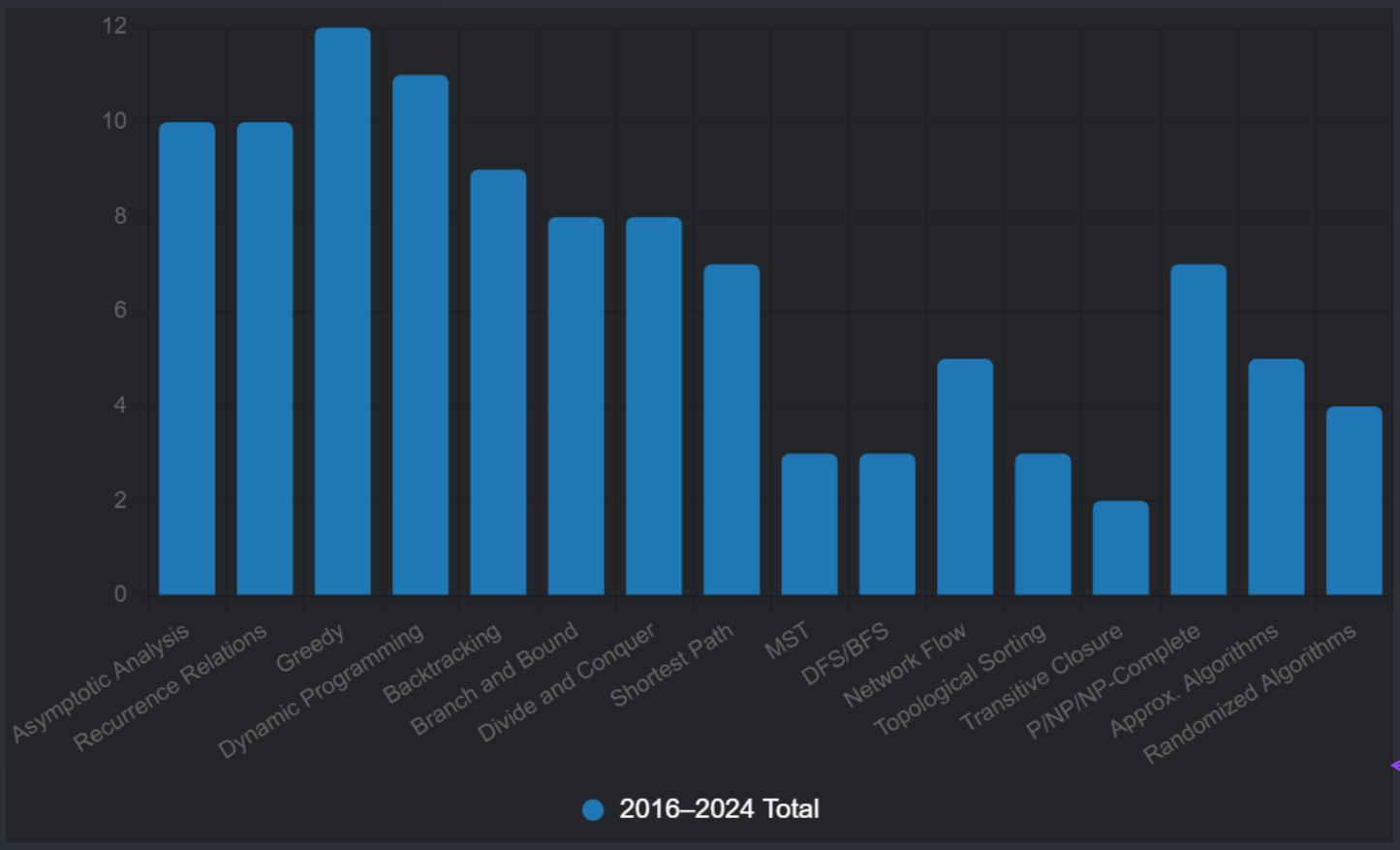
- **Network Flow:**
  - ***2019:*** Ford-Fulkerson algorithm (15 marks).
  - ***2020:*** Network Flow algorithm (15 marks).
  - ***2023 May:*** Maximum flow in directed graph (7 marks).
  - ***2023 Dec:*** Edmonds-Karp algorithm (10 marks).
  - ***2024 May:*** Ford-Fulkerson algorithm (10 marks).
  - *Topic*: Module 3 – Network Flow Algorithm.
  - *Marks*: 57 total.
- **Transitive Closure:**
  - ***2019:*** Define (1.5 marks).
  - ***2020:*** Explain (1.5 marks).
  - *Topic*: Module 3 – Transitive Closure.
  - *Marks*: 3 total.
- **Approximation Algorithms:**
  - ***2019:*** TSP (15 marks).
  - ***2020:*** General explanation (15 marks).
  - ***2023 May:*** Importance (1.5 marks).
  - ***2023 Dec:*** Vertex-Cover algorithm (8 marks).
  - ***2024 May:*** Utility with example (7 marks).
  - *Topic*: Module 5 – Approximation Algorithms.
  - *Marks*: 46.5 total.

- **Latest Questions (2024 May-specific):**
  - Recurrence relation for n-ary search (Module 1, 1.5 marks).
  - Stable sorting techniques (Module 2, 1.5 marks).
  - Optimal Merge Patterns (Module 2, 1.5 marks).
  - Dominance rule (Merge–Purge) (Module 2, 7.5 marks).

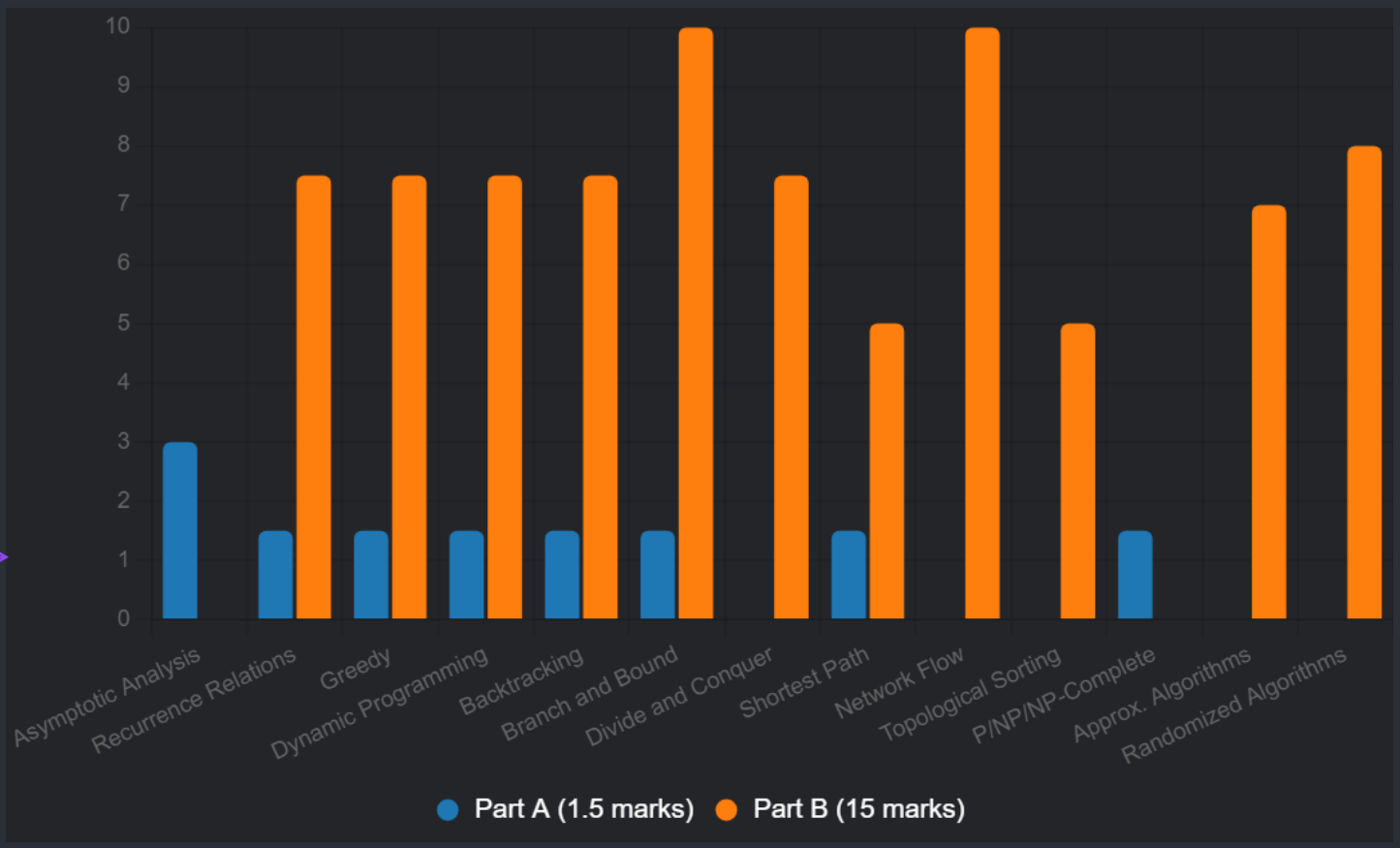
Visual Data Insights



Bar Chart: Questions per topic across 2016, 2017, 2019, 2020, 2023 May, 2023 Dec, 2024 May.



Heatmap Chart: Question frequency by topic across years.



Stacked Bar Chart: Marks distribution across topics for 2024 May.

Original Data

Click here to check the original repository

YMCA UNIVERSITY OF SCIENCE & TECHNOLOGY, FARIDABAD
B.TECH EXAM Vth SEMESTER (UNDER CBS), DEC 2016
ANALYSIS & DESIGN OF ALGORITHMS CE-307

(71)

Time: 3 hrs **M.Marks:60**

Note: Part-I is compulsory.
In Part-II, attempt 4 questions out of 6.

Part-I

Q. No. 1

a. Solve the following recurrence relation:

$$T(n)= T(n^{1/2}) + c \text{ for } n>1$$

b. Write the general algorithm for Divide and Conquer method.

c. Differentiate between upper bound and lower bound of time complexity, discuss with example.

d. What are the two way in which a graph can be represented, discuss.

e. Discuss the time complexity of Prim's method.

f. For which type of problems Backtracking approach is generally used, discuss.

g. Discuss the time complexity of All Pair Shortest Path problem.

h. Differentiate between P and NP-problems.

i. Which approach among Greedy and Dynamic method is better and why?

j. Compare the time complexity of Single Source Shortest Path problem if solved by dynamic method and Greedy method.

(2*10)

Part-II

Q. No. 2 Discuss Quick Sort technique can be used for sorting a list. How much time is required for the various steps involved in the sort? Analyze the time complexity of algorithm in best, average and worst cases. (10)

Q. No. 3 What is the solution generated by the Job sequencing with deadlines algorithm when n=7, p= (3, 5, 20, 18, 1, 6, 30) and d= (1, 3, 4, 3, 2, 1, 2)? (10)

Q. No. 4 Write the recursive backtracking algorithm for sum of subset problem and also solve the following:-

W= {5, 7, 10, 12, 15, 18, 20) and m=35

If the given set W is arranged in ascending order or descending order then is there any noticeable difference in computing time? (10)

Q.No. 5 Consider five items along with their respective weights and profits :

W_i = (5, 10, 20, 30, 40)

P_i = (30, 20, 100, 90, 160)

The Knapsack has capacity, m = 60, Find out the solution to the fractional Knapsack problem using Branch and Bound method. (10)

Q. No. 6 Discuss any two of the following problems and analyze them: (2*5=10)

a. 0/1 Knapsack Problem (using dynamic programming)
b. Hamiltonian problem (using backtracking)
c. NP-Hard and NP-Complete

Q. No. 7 Differentiate between Deterministic and Non-Deterministic algorithms. Write a Deterministic algorithm to sort an array and convert the same into a Non-Deterministic algorithm. (10)

YMCA UNIVERSITY OF SCIENCE AND TECHNOLOGY, FARIDABAD

B.TECH EXAMINATION (Under CBS)

ANALYSIS AND DESIGN OF ALGORITHMS (CE-307)

Time-3 hrs **M.Marks-60**

Note:-Part-I is compulsory.

In part-II, attempt 4 questions out of 6.

Part-I

Q.No. 1

a. Write the recurrence relation for quick sort and compute the complexity for best and worst case.

b. Describe the significance of asymptotic notation.

c. Explain Collapsing Find operations in sets.

d. Define NP-Complete and NP-Hard problems.

e. Differentiate Single source shortest path and All-pair shortest path approach.

f. Describe the general LC-Search function.

h. Explain implicit and explicit constraints for n-Queen problem.

i. Differentiate binary search tree and optimal binary search tree with an example.

j. Write a non-deterministic algorithm to sort an unsorted array.

(2*10=20)

Part-II

Q.No. 2 (a) Solve the following recurrence relations using master method:-

(i) $T(n) = T(n-5) + 1/(n+1)$

(iii) $T(n) = 2T(n/2) + n*\log n$ (4)

Q.No. 2 (b) Describe general greedy algorithm. Solve the following knapsack problem using greedy algorithm technique: m=30, (w1, w2, w3, w4) = (10, 15, 6, 9), (p1, p2, p3, p4) = (2, 5, 8, 1) (6)

Q.No.3 Write the algorithm for merge-sort and compute its time complexity. Also sort the following array using merge-sort:- (10)

A= {56, 10, 23, 6, 34, 21, 45}

Q.No.4 (a) Schedule the following jobs in an optimized way: n=7,

(p1,p2,p3,.....p7)=(3,5,20,18,1,6,30), (d1, d2, d3,.....d7)=(1,3,4,3,2,1,2). (7)

(b) Explain Principle of Optimality. (3)

Q.No.5 Write the algorithm to draw an OBST. Apply dynamic programming to draw an OBST using the following data:

	0	1	2	3	4
P(1:4)		5	20	10	5
Q(0:4)	20	10	20	5	5

(10)

Q.No.6 Write the general algorithm for iterative and recursive backtracking and also write the algorithm for graph coloring problem. (10)

Q.No.7 Solve the following travelling salesman problem using branch and bound algorithm design technique.

∞	11	10	9	6
8	∞	7	3	4
8	4	∞	4	8
11	10	5	∞	5
6	9	5	5	∞

(10)

Roll No. .....

Total Pages : 3

301404

May, 2019

B.TECH. (CE/CSE/IT) - IV SEMESTER

Design & Analysis of Algorithms (PCC-CS-404)

Time : 3 Hours]

[Max. Marks : 75

Instructions :

1. *It is compulsory to answer all the questions (1.5 marks each) of Part-A in short.*
2. *Answer any four questions from Part-B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

PART–A

1. (a) Solve the following recurrence relation:

$$T(n) = T(n^{1/2}) + c.$$ (1.5)

(b) Differentiate between O-notation and Omega-notation with the help of appropriate example. (1.5)

(c) How arrays can be used to store a tree? Discuss with example. (1.5)

(d) For which type of problems Divide & Conquer approach is generally used, discuss. (1.5)

(e) Differentiate between Depth First Search and Breadth First Search. (1.5)

(f) What is Transitive Closure of a graph? (1.5)

(g) What do you mean by reducibility? (1.5)

(h) Which approach among Backtracking and Branch & Bound is better and why? (1.5)

(i) Discuss the time complexity of Kruskal's algorithm. (1.5)

(j) Why Randomized algorithms are used? Discuss with example. (1.5)

PART–B

2. Design a divide and conquer algorithm to find the maximum and minimum of an array A of n elements, and prove that the algorithm makes at most 3n/2 element-to-element comparisons. (15)

3. (a) Differentiate between Deterministic and Non-Deterministic algorithms. Write a Deterministic algorithm to search an element in the array and convert the same into a Non-Deterministic algorithm. (7)

(b) Differentiate between Greedy and Dynamic method to solve the problems. Write and explain All Pair Shortest Path algorithm to find the Shortest Paths in a graph with example and derive its time complexity. (8)

4. Find all the feasible solutions to the Knapsack instance n = 7, m = 15,

(P_i) = (10, 5, 15, 7, 6, 18, 3)

(W_i) = (2, 3, 5, 7, 1, 4, 1).

Find the optimal solution for the same by using Greedy method. (15)

5. (a) What do you mean by Network Flow? Write Ford-Fulkerson algorithm with suitable example. (5)

(b) Differentiate between Backtracking and Branch & Bound with the help of suitable examples. Discuss the Hamiltonian Cycle problem and write its algorithm. (10)

6. (a) Why Topological sorting is used for Directed Acyclic Graph (DAG), justify with suitable example. (7)

(b) What are approximation algorithm and why they are used? Solve Travelling Salesman Problem by using approximation algorithm. (5)

7. (a) Discuss P, NP, NP-Complete and NP-Hard Problems with suitable examples. (10)

(b) What is Satisfibility? Discuss CNF-Satisfibility. (5)

October, 2020

B.Tech. (CE/CSE/IT)-IV SEMESTER

Design & Analysis of Algorithms (PCC-CS-404)

Time : 3 Hours] [Max. Marks : 75

Instructions :

1. *It is compulsory to answer all the questions (1.5 marks each) of Part -A in short.*
2. *Answer any four questions from Part -B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

PART - A

1. (a) Write the recurrence relation for ternary search and also solve it. (1.5)

   (b) Differentiate big 'O' and small 'o' asymptotic notations. (1.5)

   (c) If an array is sorted in decreasing order then which sorting runs in minimal complexity? Justify. (1.5)

   (d) Explain Transitive Closure in graph. (1.5)

   (e) Differentiate Binary tree, Binary Search Tree and Optimal Binary Search Tree. (1.5)

[P.T.O.

(f) Merge the following files optimally :

(1, 3, 2, 5, 3, 4, 6, and 8)

(g) Describe principle of optimality. (1.5)

(h) Explain explicit and implicit constraints for Hamiltonian cycle. (1.5)

(i) Describe Least Cost Search function in brief. (1.5)

(j) Differentiate NP-Hard and NP-Complete problems. (1.5)

PART - B

2. (a) Solve the following recurrence relations:-

(i) $T(n) = T(n-1) + n$

(ii) $T(n) = T(\sqrt{n}) + 1$

(iii) $T(n) = 3T(n/9) + n^3$

(iv) $T(n) = T(n/3) + T(2n/3) + n$ (10)

(b) Solve the knapsack problem (0/1 and Fractional) using Greedy Method: m = 30, (w1, w2, w3, w4) = (10, 15, 6, 9), (p1, p2, p3, p4) = (2, 5, 8, 1). (5)

3. (a) Define Merge-Purge rule with an appropriate example. (5)

(b) Write the algorithm for Quick-sort and compute its time complexity. Also sort the following array using Quick-sort:-

A = {10, 23, 6, 34, 21, 76} (10)

4. Solve the following travelling salesman problem using branch and bound:

0	7	3	12	8
3	0	6	14	9
5	8	0	6	18
9	3	5	0	11
18	14	9	8	0

(15)

5. (a) Write the backtracking algorithm for n-queen problem. Find a solution to place 4 queens on a 4*4 chess-board. (5)

(b) Write Network Flow algorithm and explain with appropriate example. (10)

6. (a) Explain Approximation algorithms in detail. (10)

(b) Define Strassen's Matrix Multiplication in brief. (5)

7. Define Cook's theorem and explain reduction of NP-Hard problems into NP-Complete problem with suitable example. (15)

Roll No. .....

Total Pages : 5

003404

May 2023

B.Tech. - IV SEMESTER

Design & Analysis of Algorithms (PCC-CS-404)

Time : 3 Hours]

[Max. Marks : 75

Instructions :

1. *It is compulsory to answer all the questions (1.5 marks each) of Part-A in short.*
2. *Answer any four questions from Part-B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

PART–A

1. (a) What is the time complexity of the following code? Justify your answer.

```
int i, j, k = 0;
  for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
      k = k + n/2;
    }}
```

(1.5)

(b) Sort the following functions in the decreasing order of their asymptotic (big-0) complexity : $f_1(n) = n^{\sqrt{n}}$, $f_2(n) = 2^n$, $f_3(n) = (1.000001)^n$, $f_4(n) = n^{10} * 2^{(n/2)}$. (1.5)

(c) Differentiate Greedy algorithm and Dynamic programming. (1.5)

(d) State Job Sequencing with Deadline Problem. Write down time and space complexity if problem solved by Greedy approach. (1.5)

(e) Define Principle of Optimality with suitable example. (1.5)

(f) Draw state space tree of 4-Queens problem. (1.5)

(g) Why does Dijkstra's algorithm fail on negative weights? (1.5)

(h) Draw binary search trees for the given set of keys and their corresponding frequencies and find the Optimal among them. keys[ ] = {10, 12, 20}, freq[ ] = {34, 8, 50} (1.5)

(i) Explain Least Cost Search function for branch and bound algorithm design technique. (1.5)

(j) What is the importance of approximation algorithm? (1.5)

PART–B

2. (a) Solve the following recurrence relation.

(i) $T(n) = 2T(n/2) + \log n$ with $T(1)=1$

(ii) $T(n) = 2T(\sqrt{n}) + 1$ if $n > 2$ and $T(n) = 2$ if $0<n<=2$. (10)

(b) How are time and space trade-offs used to optimize the performance of an algorithm? Provide an example of an algorithm that optimizes time at the expense of space and vice-versa. (5)

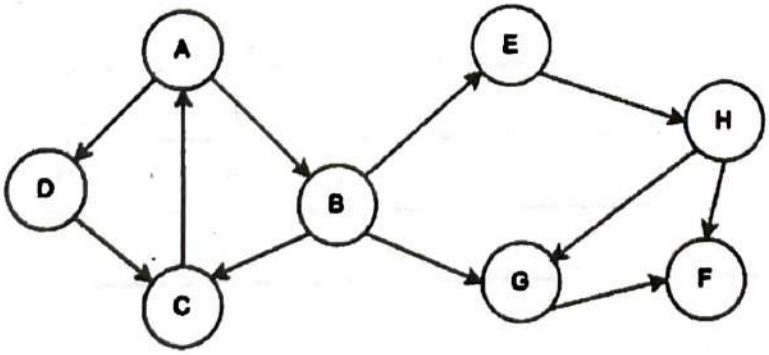
3. (a) Give a dynamic-programming solution to the 0-1 knapsack problem that runs in 0(n/W) time, where n is the number of items and W is the maximum weight of items that the thief can put in his knapsack.

Consider the weights and values of items listed below. The task is to pick a subset of these items such that their total weight is no more than 5 Kgs and o their total value is rnaximized.

Item No.	Weight (Kg)	Values (Rs.)
1	2	3
2	3	4
3	4	5
4	5	6

(b) Consider the given graph. In what order will the nodes be visited using a Breadth First Search and Depth First Search? (Assume starting vertex A) (5)

Note: If a node has multiple neighbors then select which is alphabetically near to node.



4. (a) The N-Queen problem is a classic problem in computer science, where the goal is to place N queens on an N × N chessboard so that no two queens attack each other.

(i) Write a brute-force algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and explain why it is not efficient for large values of N.

(ii) Write a backtracking algorithm to solve the N-Queen problem. Analyze the time complexity of your algorithm, and compare it with the brute-force algorithm. (12)

(b) Describe the Traveling Salesman Problem and explain why it is NP-complete. (3)

5. (a) A delivery truck travels between multiple destinations. The truck starts at city A and visits cities B, C, D, and E before returning to A. Design an algorithm to find the shortest path for the truck to travel while minimizing cost. Use following distance matrix to solve the problem. (8)

	A	B	C	D	E
A	0	10	20	15	30
B	10	0	25	30	20
C	20	25	0	35	15
D	15	30	35	0	10
E	30	20	15	10	0

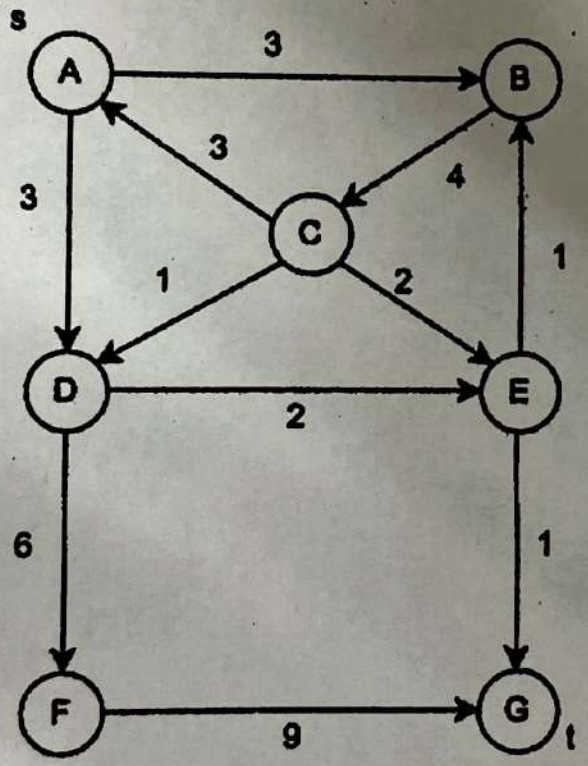
(b) Write a short note on Randomized algorithms. (7)

6. (a) Define spanning tree. Write Kruskal's algorithm for finding minimum cost spanning tree. Describe how Kruskal's algorithm is different from Prim's algorithm for finding minimum cost spanning tree. (10)

(b) What is Topological Sorting? Explain with example. (5)

7. (a) What is the relationship among P, NP, NP-Hard and NP-Complete problems? Show with the help of a diagram. (8)

(b) In a given directed graph with source s and sink t, where each edge in the graph has a non-negative capacity. Find the maximum flow that can be sent from s to t. (7)



6/12
90

Sr. No. 016502

December 2023

B.Tech (CE(DS)) ~~5th Sem~~ - V SEMESTER

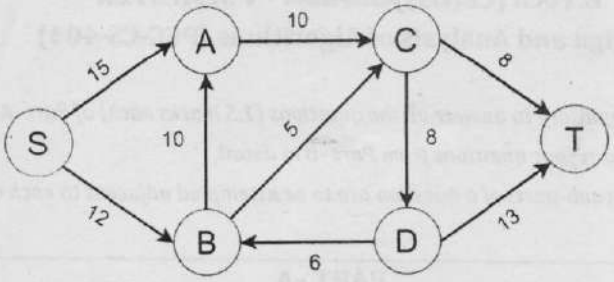
Design and Analysis of Algorithms (PCC-CS-404)

Time: 3 Hours | Max. Marks:75

Instructions:
1. *It is compulsory to answer all the questions (1.5 marks each) of Part -A in short.*
2. *Answer any four questions from Part -B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

		PART –A	
Q1	(a)	Solve the following recurrence relation: $T(n)= T(n^{1/3}) + c$ for $n>1$	(1.5)
	(b)	Differentiate between O-notation, Omega-notation and Theta with the help of appropriate example.	(1.5)
	(c)	How arrays can be used to store a tree? Discuss with example.	(1.5)
	(d)	How recurrence relations are used to find the time complexity of recursive algorithms, justify with the help of an example.	(1.5)
	(e)	Derive the time complexity for Strassen's Matrix Multiplication method.	(1.5)
	(f)	For which type of problems Greedy approach is generally used, discuss.	(1.5)
	(g)	In what cases, Dynamic Programming is better than Greedy Approach.	(1.5)
	(h)	Differentiate between Backtracking and Branch and Bound approach.	(1.5)
	(i)	What do you mean by Satisfiblitity?	(1.5)
	(j)	Why some times randomized algorithms are better than the general algorithms?	(1.5)
		PART –B	
Q2		What is Divide and Conquer approach for problem solving? Design a Divide and Conquer based algorithm to find out the kth smallest element from a given array. Also find out the time complexity for the algorithm.	(15)
Q3	(a)	Solve the following recurrence relation: $T(n)= T(1) + T(n-1) + cn$ for $n>1$	(8)
	(b)	Differentiate Breadth First Search (BFS) and Depth First Search (DFS) Graph traversal Algorithms.	(7)
Q4		Differentiate between Greedy and Dynamic method to solve the problems. Write and explain Single Source Shortest Path algorithm to find the Shortest Paths in a graph with example and derive its time complexity.	(15)
Q5	(a)	What is Backtracking? Design an algorithm to solve N-Queen problem by using Backtracking. Also specify the explicit and implicit constraints associated with N-Queen problem.	(8)

016502 / 90 / 111 / 450

	(b)	Solve the following Knapsack problem by using Branch and Bound method: N=4, P=(10, 10, 12, 18), W=(2, 4, 6, 9) and m=15	(7)
Q6	(a)	What do you mean by Network Flow? Find out the maximum network flow from the given graph using Edmond-Karp algorithm: 	(10)
	(b)	What do you mean by Randomized Algorithms? Explain Monte-Carlo and Las-Vegas Algorithms with suitable examples.	(5)
Q7	(a)	What is Approximation algorithm and where these algorithms are used? Write Approximation algorithm for Vertex-cover problem.	(8)
	(b)	Discuss P, NP, NP-Complete and NP-Hard Problems with suitable examples.	(7)

****

Sr. No. 003404

May 2024

B.Tech(CE/CE(HINDI)/IT/CSE(AIML)) - IV SEMESTER

Design & Analysis of Algorithms (PCC-CS-404)

एल्गोरिदम का डिजाईन और विश्लेषण (पीसीसी-सीएस-404)

Time(समय): 3 Hours(3 घंटे) **Max. Marks(कुल अंक) :75**

Instructions: 1. *It is compulsory to answer all the questions (1.5 marks each) of Part -A in short.*

निर्देश: **भाग-क के सभी प्रश्नों (प्रत्येक 1.5 अंक) का उत्तर संक्षेप में देना अनिवार्य है**

2. *Answer any four questions from Part -B in detail.*

भाग-ख से किन्हीं चार प्रश्नों के उत्तर विस्तार से दीजिए।

3. *Different sub-parts of a question are to be attempted adjacent to each other.*

एक प्रश्न के विभिन्न उप-भागों को एक दूसरे के निकट करने का प्रयास किया जाना है।

PART–A (भाग - क)

Q1 (a) Write the recurrence relation for n-ary search and why it is not preferred? (1.5)

एन-एरी खोज के लिए पुनरावृत्ति संबंध लिखें और इसे प्राथमिकता क्यों नहीं दी जाती है?

(b) Write the following functions in the increasing order of asymptotic complexity: (1.5)

निम्नलिखित कार्यों को स्पर्शोन्मुख जटिलता के बढ़ते क्रम में लिखिए?

$f1(n) = 2^n$ $f2(n) = n^{(3/2)}$ $f3(n) = n*\log(n)$ $f4(n) = n^{\log(n)}$

(c) Why Greedy method is not suggested for 0/1 Knapsack problem? (1.5)

0/1 नैपसेक समस्या के लिए लालची विधि का सुझाव क्यों नहीं दिया जाता है?

(d) Bellman-Ford Algorithm always find negative weight cycle in the graph? Justify your answer. (1.5)

बेलमैन-फोर्ड एल्गोरिथम हमेशा ग्राफ़ में नकारात्मक वजन चक्र ढूंढता है? अपने जवाब का औचित्य साबित करें।

(e) In which scenario a sorting technique is called stable? Also give example of 2 stable sorting techniques. (1.5)

किस परिदृश्य में छँटाई तकनीक को स्थिर कहा जाता है? दो स्थिर छँटाई तकनीकों का उदाहरण भी दें।

(f) Merge the following files optimally: (5,9,4,2,1,10) (1.5)

निम्नलिखित फ़ाइलों को इष्टतम रूप से मर्ज करें: (5,9,4,2,1,10)

(g) Explain the significance of asymptotic notations. (1.5)

स्पर्शोन्मुख संकेतन के महत्व को समझाइये।

(h) Explain explicit and implicit constraints for Hamiltonian Cycle. (1.5)

हैमिल्टनियन चक्र के लिए स्पष्ट और अंतर्निहित बाधाओं की व्याख्या करें।

(i) Describe Least Cost Search function in brief. (1.5)

(j) Differentiate P, NP-Hard and NP-Complete problems. (1.5)

पी, एनपी-हार्ड और एनपी-कम्प्लीट समस्याओं में अंतर करें।

PART-B (भाग - ख)

Q2 (a) Solve the following recurrence relations: (5)

निम्नलिखित पुनरावृत्ति संबंधों को हल करें:

(i) $T(n) = T(\sqrt{n}) + n$

(ii) $T(n) = 7T\ \text{Floor}(n/3) + n^2$

(b) Write the algorithm for divide and conquer based sorting technique which runs with same time complexity in all scenarios and also compute the time complexity of the algorithm. (10)

डिवाइड और कॉनकॉर पर आधारित सॉर्टिंग तकनीक के लिए एल्गोरिदम लिखें जो सभी परिदृश्यों में समान समय जटिलता के साथ चलता है और एल्गोरिदम की समय जटिलता की गणना भी करता है।

Q3 (a) Explain dominance rule (Merge-Purge) with an appropriate example. (5)

प्रभुत्व नियम (मर्ज-पर्ज) को उचित उदाहरण सहित समझाइये।

(b) Differentiate backtracking and dynamic programming algorithm design techniques corresponding to their application domain. Also explain the N-Queen problem in detail using backtracking. (10)

उनके एप्लिकेशन डोमेन के अनुरूप बैकट्रैकिंग और डायनेमिक प्रोग्रामिंग एल्गोरिदम डिज़ाइन तकनीकों को अलग करें। बैकट्रैकिंग का उपयोग करके एन-क्वीन समस्या को भी विस्तार से समझाएं।

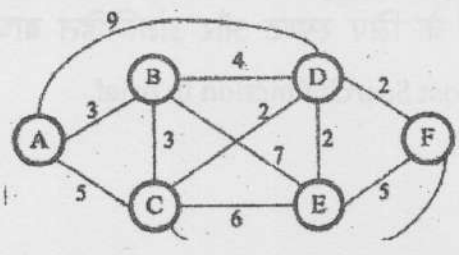
Q4 (a) Amit resides at city A and plans to visit various cities B, C and D. Compute optimal route with minimum cost to return back to A and every city he wants to visit once. (15)

अमित शहर A में रहता है और विभिन्न शहरों B, C और D का दौरा करने की योजना बना रहा है। A और प्रत्येक शहर में वापस लौटने के लिए न्यूनतम लागत के साथ इष्टतम मार्ग की गणना करें, जहां वह एक बार जाना चाहता है।

	A	B	C	D
A	0	8	4	12
B	4	0	6	14
C	6	8	0	7
D	9	4	5	0

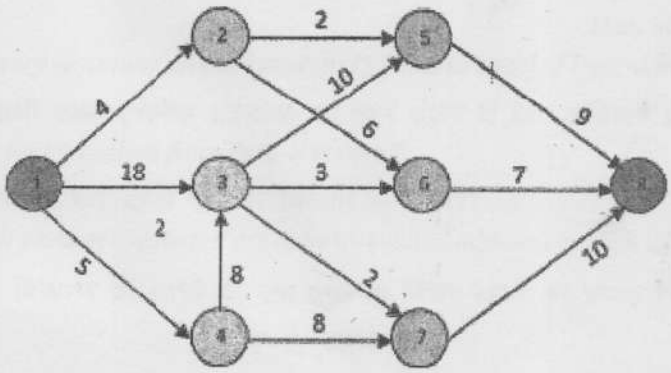
Q5 (a) Compute the Shortest path from A to F using greedy method. (5)

लालची पद्धति का उपयोग करके A से F तक के सबसे छोटे पथ की गणना करें।



(b) Write Ford Fulkerson-Network Flow algorithm and compute the maximum unit flow from node 1 to node 8. (10)

फोर्ड फुलकर्सन-नेटवर्क फ्लो एल्गोरिदम लिखें और नोड 1 से नोड 8 तक अधिकतम इकाई प्रवाह की गणना करें।



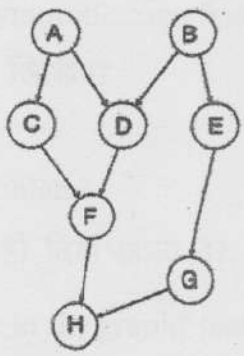
Q6 (a) Solve the knapsack problem using Branch and bound technique: (10)

ब्रांच और बाउंड तकनीक का उपयोग करके नैपसैक समस्या को हल करें

m=30, (w1, w2, w3, w4) = (18, 15, 8, 12), (p1, p2, p3, p4) = (36, 24, 8, 12)

(b) If A to H are different steps of a task and their dependency is given in the graph below. Compute topological order of events for successfully completion of task. (5)

यदि A से H किसी कार्य के अलग-अलग चरण हैं और उनकी निर्भरता नीचे ग्राफ़ में दी गई है। कार्य को सफलतापूर्वक पूरा करने के लिए घटनाओं के टोपोलॉजिकल क्रम की गणना करें।



Q7 (a) Explain any 2 types of Randomized algorithm techniques with corresponding applications and scope. (8)

संबंधित अनुप्रयोगों और दायरे के साथ किन्हीं 2 प्रकार की यादृच्छिक एल्गोरिदम तकनीकों की व्याख्या करें।

(b) Explain the utility of Approximation algorithms with appropriate example. (7)

उचित उदाहरण सहित एप्रोक्सिमेशन एल्गोरिदम की उपयोगिता समझाइए

****