Guru Nanak Dev Engineering College, Ludhiana

Department of Information Technology

Program	B.Tech.(IT)	**Semester**	6th
Subject Code	PCIT-113	**Subject Title**	Design and Analysis of Algorithms
Mid Semester Examination (MSE) No.	MSE -I (Jan-May, 2024)	**Course Coordinator**	Er. Parminder Kaur Wadhwa
Max. Marks	24	**Time Duration**	1 hour 30 minutes

Note: Attempt all questions **Roll No.** _____

Q. No.	Question	COs, RBT level	Marks
Q.1	Define the principle of optimality.	CO2, L1	2
Q.2	Formulate the θ – notation of $f(n) = 2n^2 + 3n + 1$	CO1, L6	2
Q.3	Illustrate the performance of Quicksort algorithm for its average and worst cases.	CO1, L3	4
Q.4	Demonstrate the application of dynamic-programming approach to solve all-pairs shortest paths problem.	CO2, L3	4
Q.5	Compare the efficiency of Bellman and Ford Algorithm with Dijkstra's algorithm to solve single-source paths problem in a graph.	CO3, L4	4
Q.6	Design a greedy-algorithmic approach in order to earn *maximum profit* for the following fractional Knapsack instance:- Number of elements = n = 4 Maximum Capacity of Knapsack = m = 15 Profits = (p_1, p_2, p_3, p_4) = (10, 16, 20, 18) Weights = (w_1, w_2, w_3, w_4) = (5, 2, 4, 6)	CO2, L6	8

Course Outcomes (CO) *Students will be able to*

1.	Develop an understanding of time and space complexities of an algorithm.
2.	Explore basic algorithm design techniques like divide and conquer, greedy, dynamic programming.
3.	Explore the various problem solving techniques related to graphs.
4.	Solve problems related to strings by applying various algorithms.
5.	Identify the relationship between P, NP, NP-hard and NP-complete problems.
6.	Understand the importance of approximation algorithms.

RBT Classification	**Lower Order Thinking Levels (LOTS)**			**Higher Order Thinking Levels (HOTS)**		
RBT Level No.	L1	L2	L3	L4	L5	L6
RBT Level Name	Remembering	Understanding	Applying	Analyzing	Evaluating	Creating

Guru Nanak Dev Engineering College, Ludhiana

Department of Information Technology

Program	B.Tech.(IT)	**Semester**	6th
Subject Code	PCIT-113	**Subject Title**	Design and Analysis of Algorithms
Mid Semester Examination (MSE) No.	1st MSE (Jan-June, 2023)	**Course Coordinator**	Er. Parminder Kaur Wadhwa
Max. Marks	24	**Time Duration**	1 hour 30 minutes

Note: Attempt all questions **Student's University Roll No.** 2000024

Q. No.	Question	COs, RBT level	Marks
Q.1	Define the principle of optimality.	CO2, L1	2

Q.2	Test the efficiency of the following code snippet in terms of Big Oh notation:-
	`for (i=0; i < n; i++)`
	`{`
	`for (j=0; j< i; j++)`
	`{ statement; }`
	`}`
		CO1, L4	2

Q.3	Illustrate the average and worst cases of Quicksort algorithm in terms of time complexity.	CO2, L3	4
Q.4	Use greedy method and find an optimal solution to the Knapsack instance:- n= 7, m=15, $(p_1, p_2, \ldots, p_7) = (10, 5, 15, 7, 6, 18, 3)$ $(w_1, w_2, \ldots, w_7) = (2, 3, 5, 7, 1, 4, 1)$	CO2, L3	4
Q.5	Compare the working and performance of Prim's and Kruskal's algorithms to compute minimum cost spanning tree.	CO3, L4	4
Q.6	Defend the statement that Bellman and Ford algorithm is superior than Dijkstra's Algorithm.	CO3, L5	8

Course Outcomes (CO) *Students will be able to*

1.	Develop an understanding of time and space complexities of an algorithm.
2.	Explore basic algorithm design techniques like divide and conquer, greedy, dynamic programming.
3.	Explore the various problem solving techniques related to graphs.
4.	Solve problems related to strings by applying various algorithms.
5.	Identify the relationship between P, NP, NP-hard and NP-complete problems.
6.	Understand the importance of approximation algorithms.

RBT Classification	**Lower Order Thinking Levels (LOTS)**			**Higher Order Thinking Levels (HOTS)**		
RBT Level Number	L1	L2	L3	L4	L5	L6
RBT Level Name	Remembering	Understanding	Applying	Analyzing	Evaluating	Creating

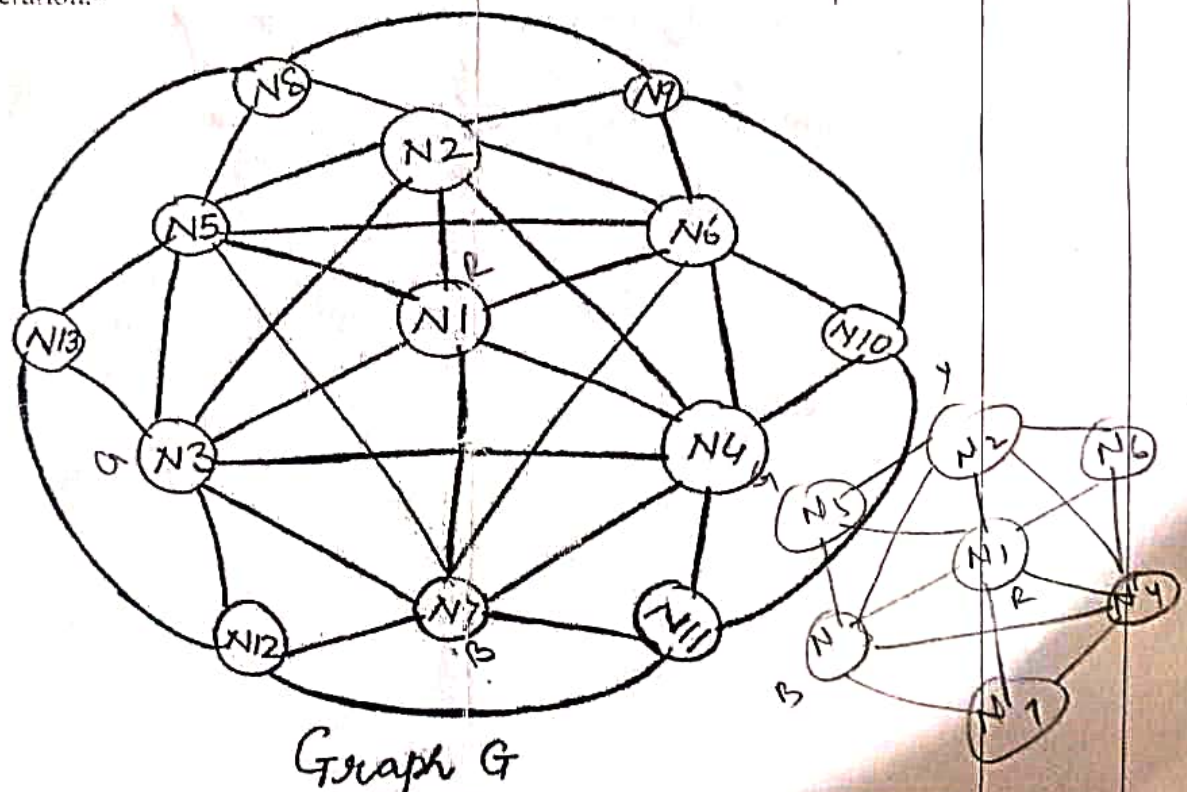
Guru Nanak Dev Engineering College, Ludhiana

Department of Information Technology

Program	B.Tech.(IT)	**Semester**	6
Subject Code	PCIT-113	**Subject Title**	Design and Analysis of Algorithms
Mid Semester Examination (MSE) No.	MSE -II (Jan-May, 2023)	**Course Coordinator**	Er. Parminder Kaur Wadhwa
Max. Marks	24	**Time Duration**	1 hour 30 minutes

Note: Attempt all questions **Roll No.** _____

Q. No.	Question	COs, RBT level	Marks
Q.1	What are ε – approximate algorithms?	CO6, L2	2
Q.2	Appraise the effectiveness of the anticipated debut of IBM Condor in 2023 in dealing with hard-optimization problems.	CO5, L5	2
Q.3	Illustrate the dynamic programming approach used by the Floyd-Warshal algorithm in order to solve all-pairs shortest paths problem.	CO2, L3	4
Q.4	Compute the chromatic number for the given Graph G by using backtracking algorithmic design technique and demonstrate the tree-generation:-	CO3, L3	4



Q.5	Assess the importance of using greedy method and relaxing the condition of $x_i = 0$ or 1 to $0 \leq x_i \leq 1$ while computing optimal solution for 0/1 Knapsack problem using a recursive backtracking algorithm.	CO2, L4	4
Q.6	Examine how Boyer-Moore algorithm is working efficiently than Boyer-Moore Horspool algorithm for the worst case time complexity.	CO4, L4	8

Course Outcomes (CO)

Students will be able to

1.	Develop an understanding of time and space complexities of an algorithm.
2.	Explore basic algorithm design techniques like divide and conquer, greedy, dynamic programming.
3.	Explore the various problem solving techniques related to graphs.
4.	Solve problems related to strings by applying various algorithms.
5.	Identify the relationship between P, NP, NP-hard and NP-complete problems.
6.	Understand the importance of approximation algorithms.

Higher Order Thinking Levels (HOTS)

Guru Nanak Dev Engineering College, Ludhiana			
Department of Information Technology			
Program	B.Tech.(IT)	**Semester**	6th
Subject Code	PCIT-113	**Subject Title**	Design and Analysis of Algorithms
Mid Semester Examination (MSE) No.	MSE -II (Jan-May, 2024)	**Course Coordinator**	Er. Parminder Kaur Wadhwa
Max. Marks	24	**Time Duration**	1 hour 30 minutes

Note: Attempt all questions **Roll No.** 2104514

Q. No.	Question	COs, RBT level	Marks
Q.1	Illustrate the concept of $f(n)$ and ε-approximate algorithms.	CO6, L3	2
Q.2	Compare the performance of Boyer-Moore algorithm and Boyer-Moore Horspool algorithm.	CO4, L4	2
Q.3	Demonstrate the working of Knuth-Morris-Pratt (KMP) algorithm.	CO4, L3	4
Q.4	Elaborate how the backtracking technique is applied to solve 0/1 Knapsack problem.	CO2, L3	4
Q.5	Design a backtracking algorithm to find all the Hamiltonian cycles in a graph.	CO3, L6	4
Q.6	Judge the correctness of the statement that "NP-hard decision problem is always NP-complete." If you judge it as correct or incorrect, give a detailed *evidence* to support your answer.	CO5, L5	8

Course Outcomes (CO) *Students will be able to*

1.	Develop an understanding of time and space complexities of an algorithm.
2.	Explore basic algorithm design techniques like divide and conquer, greedy, dynamic programming.
3.	Explore the various problem solving techniques related to graphs.
4.	Solve problems related to strings by applying various algorithms.
5.	Identify the relationship between P, NP, NP-hard and NP-complete problems.
6.	Understand the importance of approximation algorithms.

RBT Classification	**Lower Order Thinking Levels (LOTS)**			**Higher Order Thinking Levels (HOTS)**		
RBT Level No.	L1	L2	L3	L4	L5	L6
RBT Level Name	Remembering	Understanding	Applying	Analyzing	Evaluating	Creating

Please check that this question paper contains_09____ questions and _02__ printed pages within first ten minutes.

[Total No. of Questions: 09]
Uni. Roll No. 2104514

[Total No. of Pages: 02]

Program: B.Tech. (Batch 2018 onward)
Semester: 6th
Name of Subject: Design and Analysis of Algorithm
Subject Code: PCIT-113
Paper ID: 17205
Scientific calculator is Allowed

Detail of allowed codes/charts/tables etc. NIL
Time Allowed: 03 Hours **Max. Marks: 60**

NOTE:

1) Parts A and B are compulsory
2) Part-C has Two Questions Q8 and Q9. Both are compulsory, but with internal choice
3) Any missing data may be assumed appropriately

Part – A **[Marks: 02 each]**

Q1.

a) Provide a scenario where understanding asymptotic notations is crucial for designing and analyzing algorithms effectively.

b) Define dynamic programming and explain its significance in algorithm design. How does it differ from other algorithm design techniques?

c) A sorting method is said to be stable if at the end of the method, identical elements occur in the same order as in the original unsorted set. Are the merge sort and quick sort are stable sorting methods?

d) Define the chromatic number of a graph. Show that the chromatic number of a bipartite graph is always 2.

e) Contrast the concepts of NP-complete and NP-hard. How do they differ?

f) In a complete graph with 6 vertices, how many distinct Hamiltonian cycles are there?

Part – B **[Marks: 04 each]**

Q2. Define backtracking and explain its significance in algorithm design. How does backtracking differ from other algorithmic techniques?

P.T.O.

Q3. What are the characteristics of a good approximation algorithm?

Q4. What are the steps involved in implementing a greedy algorithm for the Knapsack problem?

Q5. Implement and demonstrate binary search algorithm to find the position of the element 35 in the array [5, 10, 15, 20, 25, 30, 35, 40, 45, 50].

Q6. Outline the main steps of the Rabin-Karp algorithm. Discuss the time complexity of the Rabin-Karp algorithm for searching for a substring of length m within a text of length n.

Q7. Solve the recurrence relation $a_n=a_{n-1}+n$ with initial term $a_0=4$.

Part – C **[Marks: 12 each]**

Q8. What problem does the Boyer-Moore algorithm aim to solve? How does the Boyer-Moore algorithm differ from other string matching algorithms?

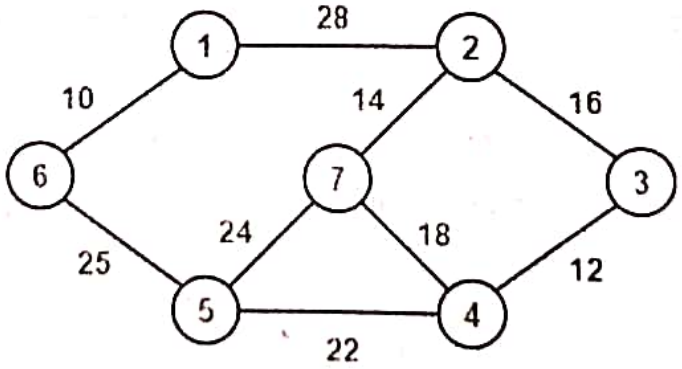
OR

What is the concept of absolute approximation in algorithm analysis? How do ε-approximations differ from absolute approximations?

Q9. Describe the quicksort algorithm and its basic steps. What is the time complexity of quicksort in the average case and worst case scenarios? Perform a quicksort on the array [12, 9, 3, 7, 14, 11, 6, 2]. Show each step and the sorted array.

OR

Write down Kruskal Algorithm to find minimal spanning tree. Apply and demonstrate the steps of the Kruskal algorithm to find the minimal spanning tree for the following graph.



****

Please check that this question paper contains ___9___ questions and ___2___ printed pages within first ten

[Total No. of Questions: 09] [Total No. of Pages: ..2....]

Uni. Roll No. .....

Program: B.Tech. (Batch 2018 onward)
Semester: 6th
Name of Subject: Design and Analysis of Algorithm
Subject Code: PCIT-113
Paper ID: 17205

Time Allowed: 03 Hours **Max. Marks: 60**

NOTE:

1) Parts A and B are compulsory
2) Part-C has Two Questions Q8 and Q9. Both are compulsory, but with internal choice
3) Any missing data may be assumed appropriately

Part – A **[Marks: 02 each]**

Q1.

a) Describe the characteristics of algorithm.
b) Which Big-O notation has the worst time complexity?
c) What are limitations of Dijkstra's Algorithm?
d) Define the principle of Optimality.
e) Differentiate NP-hard and NP-complete problems with suitable example.
f) Explain Graph-coloring problem.

Part – B **[Marks: 04 each]**

Q2. Illustrate the dynamic programming approach used by Floyd-Warshal Algorithm in order to solve all pairs shortest path algorithm.

Q3. Examine the best and worst cases of Quicksort Algorithm by performing its detailed time complexity analysis.

Q4. Differentiate Divide-and- Conquer and Greedy Method.

P.T.O.

Q5. For the given set of items and the knapsack capacity of 10 kg, find the subset of the items to be added in the knapsack such that the profit is maximum.

Items	1	2	3	4	5
Weights (in kg)	3	3	2	5	1
Profits	10	15	10	12	8

Q6. Solve the following Recurrence Relation: $T(n) = T(n-1) + \log n$.

Q7. Demonstrate how backtracking can be used to solve 8-Queens Problem.

Part – C

[Marks: 12 each]

Q8. Describe the working and performance of Prim's Algorithm to compute Minimum co Spanning tree.

OR

Explain the working of Rabin-Karp Algorithm and Knuth-Morris Pratt Algorithm.

Q9. Solve Travelling Salesperson Problem using Branch and Bound Algorithms.

OR

Consider the sum-of-subset problem, n = 4, Sum = 13, and $w_1 = 3$, $w_2 = 4$, $w_3 = 5$ a $w_4 = 6$. Find a solution to the problem using backtracking. Show the state-space tr leading to the solution. Also, number the nodes in the tree in the order of recursi calls.