Design & Analysis of Algorithms:-

Question-Bank for MST-II

Subject Incharges:-
Prof. Parminder Kaur Wadhwa
Dr. Manjot Kaur

MST-II Syllabus for DAA (PCIT-113)

Backtracking, String Matching Algorithms, NP-completeness and Approximation Algorithms

Question Bank:-

Q.1. Describe in detail general backtracking algorithm. [CO2, L2]

Q.2. Demonstrate how backtracking can be used to solve n-Queens' problem. [CO2, L3]

Q.3. Illustrate the bounding functions involved in solving sum of subsets problem using backtracking technique. [CO2, L3]

Q.4. Solve sum of subsets problem using backtracking if the elements, n=4, are {11, 13, 24, 7} and the required sum is m = 31. [CO2, L3]

Q.5. Demonstrate the use of backtracking technique to solve graph coloring problem for the following graph:-

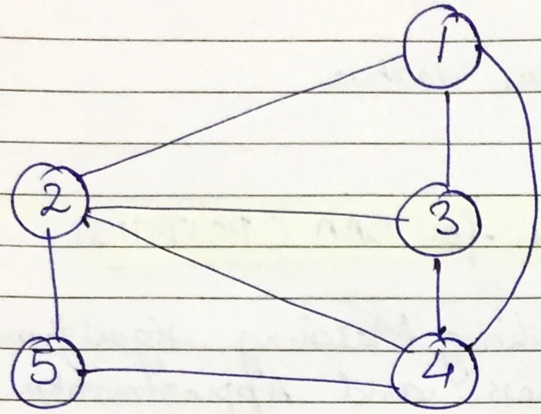

Fig:- Graph G

$n = 5$ (no. of vertices of graph G)

Compute the required number of colors to color the graph using backtracking technique.

[CO3, L3]

Q.6. Design a recursive backtracking algorithm to find all the Hamiltonian cycles in a graph.

[CO3, L6]

Q.7. Appraise the importance of using greedy method and relaxing the condition of $x_i = 0$ or 1 to $0 \le x_i \le 1$ when computing optimal solution for 0/1 Knapsack problem in a recursive backtracking algorithm.

[CO2, L4]

Q.8. Examine the various steps involved in solving 0/1 Knapsack by using backtracking algorithm design technique for the following values:-

Item	x_1	x_2	x_3	x_4
profit	45	30	45	10
weight	3	5	9	5

[CO2, L4]

Q.9. Illustrate the working of Rabin-Karp algorithm for string matching.

[CO4, L3]

Q.10. Examine how Boyer-Moore algorithm is working efficiently as string matching algorithm.

[CO4, L4]

Q.11. Demonstrate the working of Knuth-Morris Pratt (KMP) algorithm.

[CO4, L3]

Q.12. Compare the Brute Force Pattern Matching Algorithm and Knuth-Morris Pratt Algorithm.

[CO4, L4]

Q.13. Define satisfiability problem.

[CO5, L1]

Q.14. Differentiate between implicit and explicit constraints for a problem.

[CO2, L4]

Q.15. Interpret the concept of a non-deterministic algorithm.

[CO5, L3]

Q.16. Examine the relationship between P and NP problem classes.

[CO5, L4]

Q.17. Define Conjunctive Normal form (CNF).

[CO5, L1]

Q.18. Define Disjunctive Normal Form (DNF).

[CO5, L1]

Q.19. Define CNF - Satisfiability.

[CO5, L1]

Q.20. Explain Cook's Theorem.

[CO5, L1]

Date
Page

Q.21. If two problems are given L_1 and L_2. Illustrate when you can say that $L_1 \propto L_2$.

[CO5, L3]

Q.22. Examine the relationship ~~bet~~ among P, NP, NP-hard and NP-complete problem classes.

[CO5, L4]

Q.23. Support the statement that an optimization problem cannot be NP-complete whereas a decision problem can be NP-complete.

[CO5, L5]

Q.24. Defend the statement that there are NP-hard decision problems that are not NP-complete.

[CO5, L5]

Q.25. Judge the correctness of the statement that all NP-complete problems are NP-hard, but some NP-hard problems are not known to be NP-complete.

[CO5, L5]

Q.26. Write an example of a NP-hard decision problem that is not NP-complete.

[CO5, L3]

Q.27. Evaluate the Validity of the statement that P problem classes are the proper subset of NP problem classes.

[CO5, L5]

Q.28. Appraise the importance of reducing an optimization problem to its corresponding decision problem so that it can become NP-complete.

[CO5, L4]

Q.29. Illustrate the importance of approximation algorithms.

[CO6, L3]

Q.30. Describe the various categories of approximation algorithms.

[CO6, L2]

Q.31. Discuss Absolute approximation algorithms.

[CO6, L2]

Q.32. Discuss ϵ- approximation algorithms.

[CO6, L2]

Q.33. Discuss $f(n)$ - approximate algorithms.

[CO6, L2]

Q.34. Support the statement that, if we are to produce an algorithm of low polynomial complexity to solve an NP-hard optimization problem, then it is necessary to ~~th~~ relax the meaning of "solve".

[CO6, L5]

Q.35. Illustrate two relaxations of the meaning of "solve" that are made to solve NP-hard optimization problem with low polynomial complexity.

[CO6, L3]

Course Outcomes

Semester: 6th ; Class/Year: 3rd Year Information Technology

Subject: Design and Analysis of Algorithms

Subject Code: PCIT-113

Course Outcomes

Upon completion of the course students will be able to:

CO	Course Outcome
CO1	Develop an understanding of time and space complexities of an algorithm.
CO2	Explore basic algorithm design techniques like divide and conquer, greedy, dynamic programming.
CO3	Explore the various problem solving techniques related to graphs.
CO4	Solve problems related to strings by applying various algorithms.
CO5	Identify the relationship between P, NP, NP-hard and NP-complete problems.
CO6	Understand the importance of approximation algorithms.

Revised Blooms Taxonomy (RBT) Classification

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