Code No: 45015 **R07** **Set No - 1**

III B.Tech I Semester Regular Examinations,Nov/Dec 2009
DESIGN AND ANALYSIS OF ALGORITHMS
Computer Science And Engineering

Time: 3 hours **Max Marks: 80**

Answer any FIVE Questions
All Questions carry equal marks

⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Differentiate between Dynamic Knapsack and Branch and Bound Knapsack problem.
   (b) Compare and contrast Backtracking and Branch-and-Bound. How Branch-and-Bound method efficient in implementation than Dynamic Programming. [8+8]

2. Write an algorithm to find the intersection of given two graphs G1, G2. Also find time complexity of the algorithm. [16]

3. (a) Explain the divide and conquer strategy. How it can be useful in the problem solving.
   (b) Assuming that quick sort uses the first item in the list as the pivot item: i)Give a list of n items (for example, an array of 10 integers) representing the worst-case scenario. ii)Give a list of n items (for example, an array of 10 integers) representing in the best-case scenario. [8+8]

4. (a) Find at least two instances of the n-Queens problem that have no solutions?
   (b) Use the Backtracking algorithm for the m-Coloring problem to find all possible colorings of the graph 1 using the three colors red, green and white. Show the actions step by step. [8+8]

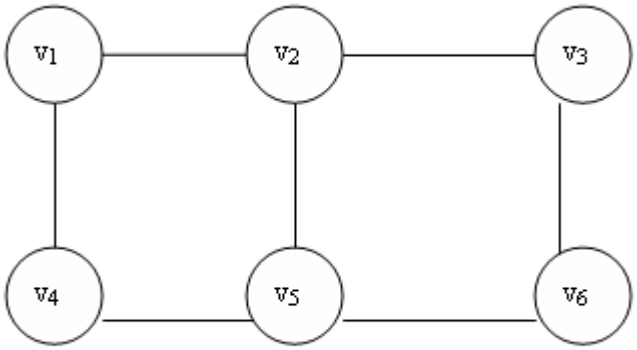

Figure 1:

5. (a) Explain how to implement Warshall's algorithm without using extra memory for storing elements of the algorithm's intermediate matrices.

(b) Give an example of a graph or a digraph with negative weights for which Floyd's algorithm does not yield the correct result. [8+8]

6. (a) Explain the Dijkstra's algorithm for single source shortest path problem with an example.

   (b) Prove that any weighted connected graph with distinct weights has exactly one minimum spanning tree. [8+8]

7. Suppose you are choosing between the following three algorithms:

   (a) Algorithm A solves problems by dividing them into five subproblems of half the size, recursively solving each subproblem, and then combining the solutions in linear time.

   (b) Algorithm B solves problems of size n by recursively solving two subproblems of size (n-1) and then combining the solutions in constant time.

   (c) Algorithm C solves problems of size n by dividing them into nine subproblems of size n=3, recursively solving each subproblem, and then combining the solutions in O(n2) time. What are the running times of each of these algorithms (in big-O notation), and which would you choose? [16]

8. (a) Show that Clique optimization problem reduces to the clique decision problem.

   (b) Obtain a non-deterministic algorithm of complexity O(n) to determine whether there is a subset of n numbers ai, $1 \leq i \leq n$, that sums to n. [8+8]

⋆ ⋆ ⋆ ⋆ ⋆