

# Design and Analysis of Algorithms: Final Exam

- (20 points) For the given graph, which of the following vertex orderings are valid depth first search (DFS) orderings for the given graph? Breadth first search (BFS) orderings ?
  - 1 8 2 7 9 5 6 4 3
  - 1 7 4 2 5 9 6 3 8
  - 1 9 6 5 2 3 7 4 8
  - 1 2 8 9 7 5 4 6 3
  - 1 7 8 2 9 4 5 6 3
  - 1 8 7 4 9 6 5 3 2
  - 1 2 5 3 8 7 4 9 6
- (30 points) Using a reduction from the Hamiltonian Path problem, prove that the Hamiltonian Circuit problem is NP-complete
- (25 points) An *independent set* in a graph is a set of vertices such that no two vertices in this set are connected to each other. The well-known NP-complete independent set problem is defined as follows: Given a graph  $G$  and an integer value  $K$ , does  $G$  have an independent set of size  $K$ ? Write pseudo-code for a backtracking solution for this problem.
- (25 points) Using the dynamic programming algorithm discussed in class for computing the optimal binary search tree, find the cost of the optimal binary search tree for four keys whose corresponding probabilities are  $p_1 = 3/8$ ,  $p_2 = 3/8$ ,  $p_3 = 1/8$  and  $p_4 = 1/8$  respectively.
- (Extra credit question, 20 points) In a weighted graph  $G$ , let  $d_{ij}$  denote the cost of the shortest path between vertices  $v_i$  and  $v_j$ . The weight of a vertex  $v_i$  in a weighted graph  $G$  is defined as

$$weight(v_i) = \sqrt{d_{i1}^2 + d_{i2}^2 + \dots + d_{in}^2}$$

The center of gravity of a graph is the vertex with the minimum weight. Design an algorithm to find the center of gravity of a given weighted graph (give pseudo code). Provide an analysis of the runtime of your algorithm. **Note:** You may use standard algorithms that you have studied this course such as binary search, quicksort etc. You may simply invoke them wherever you need to.