CS 601 - Algorithms & Complexity: Home Assignment

Total Marks - $40 = 10 \times 4$

Instructions. Please try to be brief, clear, and technically precise. Use pseudo-codes to describe the algorithms. In order to solve the problems, one may assume that the instances are in general position, unless stated otherwise. Novelty in the answer carries marks.

Q.1. Prove that the following problem is NP-Complete: given an undirected graph $G = (V, E)$ and an integer k, return a clique of size k as well as an independent set of size k, provided both exist.

Q.2. Let us denote TSP and TSP-OPT as the decision and optimization versions of the traveling salesman problem, respectively.

In the decision version (TSP), we are given a matrix of distances, and a budget b, and the goal is to output a tour that passes through all the cities and has length $\leq b$, if such a tour exists.

In the optimization version (TSP-OPT), we are given a matrix of distances, and the goal is to compute the shortest tour that passes through all the cities.

Show that if TSP can be solved in polynomial time, then so can TSP-OPT.

Q.3. Suppose two teams, A and B, are playing a match to see who is the first to win n games (for some particular n). We can suppose that A and B are equally competent, so each has a 50% chance of winning any particular game. Suppose, they have already played $i+j$ games, of which A has won i and B has won j. Give an efficient algorithm to compute the probability that A will go on to win the match. For example, if $i = n-1$ and $j = n-3$ then the probability that A will win the match is $7/8$, since it must win any of the next three games.

Q.4. Prof. S. Cooper suggests the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become position, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t.

Is this a valid method? Either prove that it works correctly or give a counterexample.