CSCI 570 Homework 6
Spring 2023

Due Date: Apr. 23, 2023 at 11:59 P.M.

1. In Linear Programming, variables are allowed to be real numbers. Consider that you are restricting variables to be only integers, keeping everything else the same. This is called Integer Programming. Integer Programming is nothing but a Linear Programming with the added constraint that variables be integers. Prove that integer programming is NP-Hard by reduction from SAT.

2. There are N cities, and there are some undirected roads connecting them, so they form an undirected graph $G = (V, E)$. You want to know, given K and M, if there exists a subset of cities of size K, and the total number of roads between these cities is larger or equal to M. Prove that the problem is NP-Complete.

   (a) Show this problem is in NP.

   (b) Show this problem is in NP-hard.

3. Consider a modified SAT problem, SAT' in which a CNF formula with m clauses and n variables $x_1, x_2, \ldots, x_n$ outputs YES if exactly $m-2$ clauses are satisfied, and NO otherwise. Prove that SAT' is NP-Complete.

   (a) Show SAT' is in NP.

   (b) Show SAT' is in NP-hard.

4. Longest Path is the problem of deciding whether a graph $G = (V, E)$ has a simple path of length greater or equal to a given number k. Prove that the Longest path Problem is NP-complete by reduction from the Hamiltonian Path problem.

5. Assume that you are given a polynomial time algorithm that decides if a directed graph contains a Hamiltonian cycle. Describe a polynomial time algorithm that given a directed graph that contains a Hamiltonian cycle, lists a sequence of vertices (in order) that form a Hamiltonian cycle.

6. Consider the maximum cut problem in an undirected network $G = (V, E)$. In this problem, we are trying to find a set of vertices $C \subset V$ that maximizes the number of edges with one endpoint in C and the other endpoint in $V \backslash C$ (note that there is no constraint on any special nodes s and t unlike in the minimum cut problem). A simple greedy algorithm would be to start with an arbitrary set C and then ask if there is a vertex $v \in V$ such that we will increase the number of edges in the cut by moving it from one side of the cut to the other. We then iterate this process until there is no such vertex. a) Show that the algorithm must terminate, and b) show that when the algorithm stops, the size of the cut is at least half of the optimum.

7. It is well-known that planar graphs are 4-colorable. However, finding a vertex cover on planar graphs is NP-hard. Design an approximation algorithm to solve the vertex cover problem on planar graph. Prove your algorithm approximation ratio.

8. Consider a well-known 0-1 Knapsack problem. Given collection of n items with integral sizes $w_1, \ldots, w_n > 0$ and values $v1, \ldots, v_n > 0$, and an integer knapsack capacity $W > 0$. The problem to find integers $w_1, \ldots, w_n > 0$ such that the total value is maximized is known to be NP-Hard. As a possible heuristic, let's try the following greedy algorithm: sort items in non-increasing order of v_i/w_i and greedily pick items in that order. Show that this approximation algorithm is pretty bad, namely it does not provide a constant approximation.