CS 501 / MCS 501 – Computer Algorithms I
Spring 2024
Problem Set 2

Lev Reyzin

Due: 2/16/24 by the beginning of class

Instructions: Atop your problem set, write your name and also write the names of all the students with whom you have collaborated on this problem set.

1. [10 pts] Let x be an extreme point of a linear program (LP) if it is a feasible solution that cannot be expressed as a convex combination of two other feasible solutions.[1] Prove that an extreme point of the LP:

$$\begin{array}{lll} \text{minimize} & \sum_{i \in V} x_i & \\ \text{subject to} & x_i + x_j \geq 1, & \forall (i,j) \in E, \\ & x_i \geq 0, & \forall i \in V, \end{array}$$

has the property that

$$\forall i \in V, \ x_i \in \{0, 1/2, 1\}\,. \tag{1}$$

2. [10 pts] Use the property in Equation 1, together with the famous result that every planar graph can be four-colored in polynomial time, to give a polynomial-time 3/2-approximation for vertex cover on planar graphs.[2]

3. [10 pts] Give an example of an instance of a Euclidean traveling salesman problem (TSP) for which both the "nearest addition heuristic" and Christophedes approximation algorithm fail to find the optimal tour. In solving this question, give the tours that both of the above algorithms would produce on your instance and explain why neither is optimal.

4. [20 pts] Here, we will design a polynomial-time algorthm for finding small vertex covers.

- **a. [5 pts]** Prove that if a graph G has a vertex cover of size k, then G has at most kn edges.
- **b. [5 pts]** Let $e = (u, v)$ be any edge of G. Prove that G has a vertex cover of size k if and only if either $G - \{u\}$ or $G - \{v\}$ has a vertex cover of size $k - 1$.
- **c. [10 pts]** Use the above observations to design a recursive algorithm for finding a vertex cover of size k (if it exists) in a graph and show that your algorithm runs in polynomial time for $k = O(\log n)$. How does this compare with a brute-force approach?

[1] More precisely, for an extreme point x, it is not the case that for $0 < \lambda < 1$ that $x = \lambda x' + (1 - \lambda)x''$ if x' and x'' are feasible solutions distinct from x.

[2] Recall that there exist polynomial-time LP solvers that are guaranteed to return only extreme points as optima.