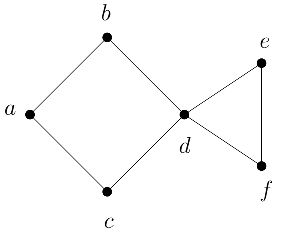
CSCI3160: Special Exercise Set 11

Prepared by Yufei Tao

Problem 1. Consider the undirected graph G below.



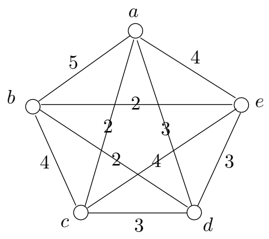
(a) What is the size of a smallest vertex cover of G?

(b) Is it possible for our vertex cover algorithm (taught in the class) to output a vertex cover of size 4?

(c) How about size 6?

Problem 2. Define "variable" and "literal" in the same way as we did for the MAX-3SAT problem. However, re-define a *clause* as the OR of an arbitrary number of literals subject to the constraint that all literals need to be defined on distinct variables. Prove: by independently setting each variable to 0 or 1 with 50% probability, we ensure that the clause should evaluate to 1 with probability at least 1/2.

Problem 3. Consider the undirected graph G below.



Use the algorithm taught in the class to find a Hamiltonian cycle that achieves an approximation ratio of 2.

Problem 4. In Step 2 of our 2-approximate algorithm for our traveling salesman problem, we need to compute a walk from the MST T computed in Step 1. Explain how to compute the walk in time proportional to the number of vertices in T.

Problem 5. Let P be a set of n points in 2D space. Define a *cycle* as a sequence of n line segments: (s_1, t_1), (s_2, t_2), ..., (s_n, t_n) where

- $s_i \in P$ and $t_i \in P$ for each $i \in [1, n]$;
- $t_i = s_{i+1}$ for all $i \in [2, n]$ and $s_1 = t_n$;
- $P = \{s_1, s_2, ..., s_n\}$;

- each (s_i, t_i) is a segment connecting points s_i and t_i.

The length of the cycle is the total length of all the n segments. Let OPT_P be the shortest length of all cycles. Design a poly(n)-time algorithm (i.e., polynomial in n) that finds a cycle with length at most $2 \cdot \mathrm{OPT}_P$.

Note: for this problem, you can assume that there is oracle that can help you calculate the distance between two points for free.