CPSC 365 / ECON 365: Algorithms Yale University

Problem Set 6

Out: April 5, 2022 Due: April 17, 2022 at 11:59 pm EST

Instructions

Write your solution to the following problems carefully. Submit the PDF of your solution via Gradescope. Please start your solution to every question on a new page. Make sure to assign the correct page in your document corresponding to each problem. We recommend writing your solution in Latex. Handwritten solutions are accepted if they are clearly legible. Write your name and SID in your answer to Problem 0; do not write them anywhere else in your solution.

Collaboration policy: You may collaborate and discuss these problems with other students. However, you must first make an honest effort to solve these problems by yourself. You may discuss hints and try to solve problems together. But you may not explicitly provide the solution to a problem to other students; nor receive the solution from them. You must write your solution independently, and make sure you understand your solution. You must list all your collaborators (anyone with whom you discussed any part of these problems). You may consult the course textbook and other references related to the class. However, you are forbidden from searching for these problems on the Internet. You must list any resources you consult (beyond the course textbook). You must also follow the Yale academic integrity policy.

0 Your Information

On the first page of your submission, include the following information (but do not include these anywhere else in your solution). Also certify that you have followed the collaboration policy. Your solution to the remainder of the problems should start on a new page.

(a) Your name.

(b) Your SID.

(c) A list your collaborators and any outside resources you consulted for this problem set. If none, write “None”.

(d) Certify that you have followed the academic integrity and collaboration policy as written above.

(e) How many hours did you spend in this problem set?

1 Minimum Spanning Tree

Let `MST` denote the following problem.

- Input: An undirected, connected, weighted graph $G = (V, E)$, and a subgraph S of G.
- Output: `Yes` if S is a minimum spanning tree (MST) in G.

Show that `MST` is in the class **NP**.

Concretely, consider a candidate solution S, which is a subgraph of G given as a subset of edges. Describe how you will do the following tasks in polynomial time, to check whether S is a MST. You may use any algorithm from class as a black box.

(a) Check that the subgraph induced by S is connected.

(b) Check that the subgraph induced by S is cycle-free.

(c) Check that the subgraph induced by S has vertex set V.

(d) Check that the subgraph induced by S is a minimum spanning tree.

2 SAT Variations

This problem illustrates how subtle variants of `SAT` have different levels of difficulty.

(a) Describe a polynomial time algorithm to solve the following problem `Prob1`.

Input: A boolean function in CNF such that each clause has exactly three literals.

Output: `Yes` if there is an assignment of the variables such that each clause has all `True` literals or all `False` literals.

(b) Show that the following problem `Prob2` is NP-hard.

Input: A boolean function in CNF such that every clause has at most three literals and every variable appears in at most three clauses.

Output: `Yes` if there is an assignment that evaluates the given function to `True`.

(*Hint:* Reduce from 3SAT.)

(c) Show that the following problem `Prob3` is NP-hard.

Input: A boolean function in CNF.

Output: `Yes` if there are *at least* three distinct satisfying assignments that evaluate the given function as `True`.

(*Hint:* Reduce from SAT.)

3 Graph Coloring

A k**-coloring** of an undirected graph $G = (V, E)$ is an assignment of a color $c(v) \in \{1, \dots, k\}$ to each vertex $v \in V$, from k possible colors $\{1, \dots, k\}$, such that for every edge $(u, w) \in E$, the endpoints have different colors: $c(u) \neq c(w)$.

Consider the k-`COLORING` problem defined as:

- Input: An undirected graph $G = (V, E)$ and an integer $k > 0$
- Output: `Yes` if a k-coloring of G exists.

You may assume that `3-COLORING` is NP-complete. Prove that `4-COLORING` is NP-complete. (That is, show that `4-COLORING` is in NP, and it is NP-hard.)