CMSC 27200 - Winter 2021 — Andrew Drucker, Lorenzo Orecchia

due tuesday, february 2 at 12 noon via gradescope

H3.Hamlet

- The assignment is due at Gradescope on Tuesday, February 2 at 12 noon.
- You can either type your homework using LaTex or scan your handwritten work. We will provide a LaTex template for each homework. If you writing by hand, please fill in the solutions in this template, inserting additional sheets as necessary. This will facilitate the grading.
- You are permitted to discuss the problems with up to 2 other students in the class (per problem); however, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Similarly, please list any other source you have used for each problem, including other textbooks or websites.
- *Show your work.* Answers without justification will be given little credit.

PROBLEM 1 (25 POINTS) *Give a polynomial-time algorithm that solves the following problem. Give a clear proof that your algorithm is correct and runs in polynomial time.*

Input: *a simple (no repeated edges or self-loops), connected, undirected graph $G = (V, E)$ with a positive edge-weight function $w : E \to \mathbb{N}^+$, with all edge weights distinct;*

Output: *a spanning tree $T \subseteq E$ of G, of second-smallest total weight.*

More precisely: T should have minimum weight among all spanning trees of G that are not *Minimum Spanning Trees for (G, w). If there are two or more such trees, you are free to output any one of them, and you don't need to decide whether such a tie can or does occur.*

Solution:

Let $K = \emptyset$
Assume $w(e_1) < w(e_2) < \ldots < w(e_m)$
Assume $i \geq 2$.
if $K \cup e_i$ is cycle-free **then**
    $K \leftarrow K \cup e_i$
end if
return $T \leftarrow K$

Kruskal's Algorithm is as Follows

Proof of polynomial time The algorithm for this proof is Kruskal's algorithm which runs on polynomial time.

Proof of correctness This proof we will use proof by contradiction. For this proof we want to show that our algorithm always outputs a tree of second-smallest total weight by proving our output is the smallest spanning tree not equal to the Minimum Spanning Tree (MST). We will work with our input and also assume that our input has at least two spanning trees where at most one in the MST. For any given $G = (V, E)$ where V is a set of vertices and E is a set of edges, let e_i be the current iteration of edges added to the partial solution of Kruskal's algorithm. First we want to show that our output does not equal the MST. Let E be the set of edges in the MST. Since all the edges have distinct weights, then E is a unique set of edges such that for any spanning tree, E', $E' = E$ if for all $e' \in E', e' = e$. Let E' be the result of our algorithm, and Let $w(e)$ be the edge with the smallest weight in E then we know $w(e) = w(e_1)$ according to Kruskal's algorithm. However, our algorithm assumes $i \geq 2$ s.t. $w(e_1)$ never enters the loop nor is included in the final E'. Thus $e \notin E'$ contradicts our assuption that $E' = E$.

Secondly we want to show that E' has the second smallest total weight. We will prove this using the same principles of the squeeze theorem.

Let X be the set of edges that make up the MST and Y be the set of edges output by our algorithm, and let Z be some set of edges such that the the total weight of Z is greater than the total weight of X and smaller than the total weight of Y. We want to show that the total weight of Y is at most the total weight of Z, $w(Y) \leq w(Z)$. We will prove this using contradiction. Let $x \in X, y \in Y$ and $z \in Z$.

The difference between e_i and e_{i+1} in Kruskal's algorithm is minimal because if e_i and e_{i+1} are both minimal since Kruskal's assumes $w(e_i) < w(e_{i+1}) < ...w(e_n)$ We know e_i and e_{i+1} are edges with weights closest to each other such that $e_{i+1} - e_i$ is minimal with every new edge added. If $\exists e' \in E$ such that $w(e') - w(e_i) < w(e_{i+1}) - w(e_i)$ then $w(e') < w(e_{i+1})$. But thmeans $e = e_{i+1}$.

Your solution goes here.

Extra Space for your solution

Extra Space for your solution

PROBLEM 2 (25 POINTS) *Solve exercise 19 in Chapter 4 (bottleneck rates) in the Kleinberg-Tardos textbook.*

Solution: Your solution goes here.

Extra Space for your solution

PROBLEM 3 (25 POINTS) *Let $G(V, E)$ be an undirected and **unweighted** graph with n nodes. Let $T_1, T_2, \ldots, T_k$ be $k = n - 1$ distinct spanning trees of G. Devise a polynomial-time algorithm that finds a spanning tree $T = (V, E_T)$ in G that contains at least one edge from each T_i. (Prove correctness and polynomial runtime.)*

Definition: two trees (or for that matter any graphs) on a set of nodes are **distinct**, if they differ in at least one edge.

Solution: Your solution goes here.

Extra Space for your solution

PROBLEM 4 (25 POINTS) *Let $G = (V, E, \{w_e\}_{e \in E})$ be an undirected graph with positive edge weights w_e indicating the lengths of the edges. Devise a polynomial-time algorithm that, given a vertex $i \in V$, computes the length of the shortest cycle containing vertex i in G. Give a proof of correctness and a running-time analysis for your algorithm. For full credit, your algorithm should run in time $O(|V|^2)$.*

Extra Space for your solution