CS 561: Homework 10

Prof. Jared Saia, University of New Mexico

Remember: You are encouraged to work on the homework in groups, but please observe the "Star Trek" rule from the syllabus.

1. **Union-Find Ranks.** Consider Union-Find with union by rank and path compression, as in lecture. Prove that every represented set X satisfies

   $$|X| \geq 2^{\text{rank}(\text{leader}(X))}.$$

   Use strong induction on $|X|$, and explicitly state the base case, inductive hypothesis, and inductive step. Account for both equal-rank and unequal-rank unions, and explain why path compression preserves the inequality.

2. **Collecting Light Edges.** Let $G = (V, E)$ be a connected, undirected, weighted graph. An edge is *light* for a cut if it has minimum weight among the edges crossing that cut; ties are allowed.

   Professor Moe collects every edge that is light for at least one cut of G and conjectures that the collected edges always form a minimum spanning tree. Give a counterexample and justify why it disproves the conjecture.

3. **A Converse of the Safe-Edge Theorem?** Let $G = (V, E)$ be a connected, undirected, weighted graph, and let $A \subseteq E$ be contained in some minimum spanning tree. Recall that a cut *respects* A if no edge of A crosses it, and that an edge $e \in E \setminus A$ is *safe for* A if $A \cup \{e\}$ is contained in some minimum spanning tree.

   Professor Matsumoto conjectures that if e is safe for A, then e is light for *every* cut that respects A and that e crosses. Prove the conjecture or give a counterexample. In a counterexample, specify G, its weights, A, e, and the cut, and verify all the hypotheses.

4. **An Edge in an MST.** Let $G = (V, E)$ be a connected, undirected, weighted graph. Prove that if an edge e belongs to some minimum spanning tree of G, then e is light for some cut of G. Your proof must allow ties among edge weights.

5. **One Half-Price Edge.** Let $G = (V, E)$ be a directed graph with nonnegative edge costs $w(e)$, and let s and t be distinct vertices. You have a coupon that lets you pay $w(e)/2$ instead of $w(e)$ for at most one edge on your route from s to t. All other edges cost their usual amounts. You may also choose not to use the coupon.

   (a) A proposed algorithm first finds a shortest s-to-t path using the original edge costs, then uses the coupon on a most expensive edge of that path. Give a counterexample showing that this algorithm need not find a minimum-cost route. Your example should have a unique shortest path under the original costs.

   (b) Design an algorithm that finds a minimum-cost route with the coupon in $O((n+m)\log n)$ time, where $n = |V|$ and $m = |E|$. Describe the algorithm, prove its correctness, and justify its running time. Explain how to return the route and identify the discounted edge, if any. If no s-to-t route exists, your algorithm should report this.