Instructions. Limited collaboration is allowed while solving problems, but you must write solutions yourself. List collaborators on your submission.

You can choose which problems to complete, but must submit at least one problem per assignment. See the course page for information about how challenge problems are graded and contribute to your homework grade. Since you don’t need to complete every problem, you are encouraged to focus your efforts on producing high-quality solutions to the problems you feel confident about. There is no benefit to guessing or writing vague answers.

If you are asked to design an algorithm, please (a) give a precise description of your algorithm using either pseudocode or language, (b) explain the intuition of the algorithm, (c) justify the correctness of the algorithm; give a proof if needed, (d) state the running time of your algorithm, (e) justify the running-time analysis.

Submissions. Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope.

1 Challenge Problems

Problem 1. Nailed it! You’re working as a behind-the-scenes baker for the Netflix show “Nailed it!” (well, just watch the show). For the next episode, you need to bake \( n \) different recipes, labeled \( r_1, r_2, \ldots, r_n \). Recipe \( r_i \) requires \( p_i \) minutes of preparation time and \( b_i \) minutes of baking time. Fortunately, the show’s kitchen has \( n \) ovens, so all recipes can bake simultaneously once they are prepared. However, there is only one of you, so you need to decide in which order to complete the preparation of each recipe.

For example: as soon as you complete preparing the first recipe, you can put it in the oven to bake and immediately begin preparing the second recipe. When you complete the second recipe, you can put it in the oven to bake whether or not the first recipe is done baking; and so on.

Let’s say that a schedule is an ordering for preparation of the recipes, and the completion time of the schedule is the earliest time at which all recipes are done baking. This is an important quantity to minimize, because recording of the episode starts soon!

Give a polynomial-time algorithm that finds a schedule with as small a completion time as possible.

Problem 2. K&T Ch 4 Ex 9. Minimum bottleneck spanning trees. Before solving this problem, read about the cycle property — Fact (4.20) on p.147 of the textbook. Here we explore a different objective for designing networks: instead of finding a spanning tree with the smallest total cost, we will explore the problem of finding a spanning tree where the most expensive edge is as small as possible.

Specifically, let \( G = (V, E) \) be a connected graph with \( n \) vertices, \( m \) edges, and positive edge costs that you may assume are all distinct. Let \( T = (V, E') \) be a spanning tree of \( G \); we define the bottleneck edge of \( T \) to be the edge of \( T \) with the greatest cost.

A spanning tree \( T \) of \( G \) is a minimum-bottleneck spanning tree if there is no spanning tree \( T' \) of \( G \) with a cheaper bottleneck edge.

1. Is every minimum-bottleneck tree a minimum spanning tree of \( G \)? Prove or give a counterexample.

2. Is every minimum spanning tree a minimum-bottleneck tree of \( G \)? Prove or give a counterexample.

Hint: use the cycle property.
Challenge Problems 3

Problem 3. Shortest Road Trip. Mary and Tom are taking a road trip from New York City to San Francisco. Because it is a 44-hour drive, Mary and Tom decide to switch off driving at each rest stop they visit. However, because Mary has a better sense of direction than Tom, she should be driving both when they depart and when they arrive (to navigate the city streets). Given a route map represented as a weighted undirected graph $G = (V, E, w)$ with positive edge weights, where vertices represent rest stops and edges represent routes between rest stops, devise an efficient algorithm to find a route (if possible) of minimum distance between New York City and San Francisco such that Mary and Tom alternate edges and Mary drives the first and last edge.

[Hint: one way to solve this problem is to construct a new graph $G'$ to represent the alternate driving]

Problem 4. K&T Chapter 4, Exercise 10. (minor variation in part (b)). Let $G = (V, E)$ be an (undirected) graph with costs $c_e \geq 0$ on the edges $e \in E$. Assume you are given a minimum-cost spanning tree $T$ in $G$. Now assume that a new edge is added to $G$, connecting two nodes $v$ and $w$ with cost $c$. You may assume that edge costs are distinct.

1. Give an efficient algorithm to test if $T$ remains the minimum-cost spanning tree with the new edge added to $G$ (but not to the tree $T$). Make your algorithm run in time $O(|E|)$. Can you do it in $O(|V|)$ time? Please note any assumptions you make about what data structure is used to represent the tree $T$ and the graph $G$

2. Suppose $T$ is no longer the minimum-cost spanning tree. Give a linear-time algorithm (time $O(|E|)$) to update the tree $T$ to a lower cost spanning tree.