Homework Six, for Fri 11/6 CSE 101

Prepare a PDF file in which your solution to each of the following problems (1–7) begins on a fresh page. Upload the file to Gradescope, using your campus email address as login. The deadline is noon on Friday.

These problems cover the following skills and concepts:

- Understanding the greedy approach to algorithm design
- Being able to obtain greedy solutions for suitable problems:
  - Finding counterexamples for bad greedy strategies
  - Justifying the correctness of a good strategy
  - Finding an efficient implementation
- Familiarity with Dijkstra's algorithm for shortest paths
- Familiarity with the greedy algorithm for set cover

---

1. *Another scheduling problem.* Textbook problem 5.32.
2. *Coloring a graph.* An undirected graph $G = (V, E)$ is *two-colorable* if it is possible to assign each vertex a color, `black` or `white`, such that no edge has endpoints of the same color.

   Give a linear-time algorithm for determining whether a graph is two-colorable. Argue that it always returns the correct answer.
3. *Dijkstra example.* Textbook problem 4.1.
4. *Non-optimality of greedy set cover.* In class, we studied a greedy algorithm for set cover. Give a small example to show that it does not always return an optimal solution.
5. Alice wants to throw a party and is deciding whom to call. She has n people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick as many people as possible, subject to a constraint: at the party, each person should have at least five other people whom they know.
   (a) Give a high-level description of an efficient algorithm that takes as input the list of n people and the list of pairs who know each other and outputs the best choice of party invitees. Argue that this scheme is correct.
   (b) Give an efficient implementation of the scheme from part (a), and analyze its running time in terms of n.
6. You are driving down a very long highway, with gas stations at mile-posts $m_1, m_2, \ldots, m_n$, where $m_1 = 0$ is your starting point and m_n is your final destination. You want to make as few gas stops as possible, but your car can only hold enough gas to cover M miles. Give an algorithm to find the minimum number of stops you need to make. Argue the correctness of the algorithm, and analyze its running time.
7. We are given a collection of intervals on the line: $I_1 = [\ell_1, u_1], \ldots, I_n = [\ell_n, u_n]$. We'd like to select a small set of points on the line, such that each interval contains at least one of the points. Give an efficient algorithm for finding the smallest possible set of such points, and make an argument for its correctness.