CS41 Lab 5

September 30 2019

In typical labs this semester, you'll be working on a number of problems in groups of 3-4 students. You will not be handing in solutions; the primary purpose of these labs is to have a low-pressure space to discuss algorithm design. However, it will be common to have some overlap between lab exercises and homework sets.

The goal of this lab session is to practice working with graphs and applying some of the design ideas we've learned for greedy and graph algorithms. Do not expect to complete all parts of all problems by the end of the lab. Work on the problem(s) that sound the most interesting to your group.

1. **Ethnographers.** (Kleinberg and Tardos, 3.12) You're helping a group of ethnographers analyze some oral history data they've collected by interviewing members of a village to learn about the lives of people who have lived there over the past two hundred years.

   From these interviews, they've learned about a set of n people (all now deceased), whom we'll denote $P_1, P_2, \ldots, P_n$. They've also collected facts about when these people lived relative to one another. Each fact has one of the following two forms:

   - for some i and j, person P_i died before person P_j was born; or
   - for some i and j, the lifespans of P_i and P_j overlapped at least partially.

   Naturally, the ethnographers are not sure that all these facts are correct; memories are not very good, and a lot of this was passed down by word of mouth. So what they'd like you to determine is whether the data they've collected is at least *internally consistent*, in the sense that there could have existed a set of people for which all the facts they've learned simultaneously hold.

   Give an efficient algorithm to do this: either it should propose dates of birth and death for each of the n people so that all the facts hold true, or it should report (correctly) that no such dates can exist—that is, the facts collected by the ethnographers are not internally consistent.

2. **Bus Transfers.** In the midst of a long backpacking trip, you've arrived in a new city that you're visiting for only one day. Starting from the train station at the southeast corner of the city, you need to reach your hostel at the city's northern boundary by the end of the day. Along the way, you want to visit as many tourist attractions as you can, but on the cheap. Luckily, the city has lots of good bus routes, and lets you transfer to another bus for free as long as it's going in the same direction.

   We can model this problem using a directed acyclic graph, where the nodes represent tourist attractions, and the edges correspond to northbound bus routes. Design and analyze a polynomial time algorithm to determine how many sites you can visit. Your algorithm should take a DAG G, a start vertex s, and a destination vertex t, and should return the maximum number of nodes that can be visited along a path $s \to t$.

3. **Making change with coins.** Consider the problem of making change for n cents out of the fewest number of coins. Assume that n and the coin values are positive integers (cents).

(a) Describe a greedy algorithm to solve the problem using the US coin denominations of quarters (25), dimes (10), nickels (5), and pennies (1). Prove your algorithm is optimal.

(b) Suppose the country of Algorithmland uses denominations that are powers of c for some integer c. This country uses $k+1$ denominations of $c^0, c^1, \ldots, c^k$. Show that your greedy algorithm works in Algorithmland as well.