Due: 11:59pm Friday, Nov 16

Lab 2

Fall 2018, Bhakta
CS 315, Algorithms

Overview In this lab, you will solve a problem using dynamic programming. You may write your assignment in any programming language that you choose, but clear it with me if you aren't planning to use C, C++, Java, Python, or MIPS. You may work in pairs, and only per person per group needs to submit the code on blackboard. Make sure to include both participants in the comments of anything you submit or in a separate text file in your submission. Your problem should read from standard-in and write to standard-out.

Problem: The In-order Traveling Salesfriend Problem You have to visit n cities $c_0, c_1, \ldots c_{n-1}$ to give sales presentations, all scheduled one after the other. You have a graph that represents n cities $c_0 \ldots c_{n-1}$ and the cost to fly from city to city. You may assume that every city has a direct flight to every other city, but some flights are more expensive than others. Also, being a successful traveling salesperson, you only take direct flights.

Since you must visit these cities *in order*, the only thing for you by yourself to do is to start at c_0, then fly to c_1, then c_2, etc. Luckily, you have a traveling salesfriend! You and your friend are going to divide the cities between yourselves. Each city must still be visited in order, but you could potentially save a lot of money.

For example, if you had a schedule like:

LA, Sacramento, NYC, Seattle, DC, San Diego, Richmond, Portland, SF, Miami

it would be very expensive to fly back and forth constantly by yourself.

But with a friend, you could potentially save a lot on airfare by splitting up the cities by coast.

You: LA, Sacramento, Seattle, San Diego, Portland, SF You: NYC, DC, Richmond, Miami

You and your friend can start and stop at any city without cost, and divide up the cities however you prefer, *as long as the cities are visited in the right order*. Without loss of generality, assume that you start at city c_0, and your friend starts anywhere.

Decide how you should minimize the cost of airfare.

Input Format Line 1: One integer: n
Lines 2-n+1: an nxn grid of integers, representing the adjacency matrix of the graph of cities, where $A[i][j]$ is the cost of the direct flight from city i to city j.

Output Format Line 1: The minimum cost that you can acheive with the two of you splitting the cities. Line 2: The list of cities that you should take. Line 3: The list of cities that your friend should take.

Solution Who worked on this submission?

Solution:

Consider the solution that involves checking through all the possible routes, and choosing the one of minimal cost. Exactly how many such routes are there¿

Solution:

Explain your construction: What are the subproblems in your recursion?

Solution:

Explain your construction: What is the recursive relationship between these problems?

Solution:

Explain your construction: What is the asymptotic runtime of your code?

Solution:

Remember to print

Your Assignment Solve the above problem using dynamic programming. First come up with the subproblems and recursive solution to the problem. Then implement this algorithm iteratively, and submit your working code as well as this pdf on blackboard.

You should test your code on more than the input given. If you want to construct an input and have me verify my "correct answer" for testing purposes, I will be happy to do that on Piazza or in my office.

Bonus (Up to 25 points)

(5 points each) Create an account on the ICPC programming contest archive. Show me that you've submitted working code for any problem that you solve using dynamic programming, up to 30 bonus points total.