CS325: Analysis of Algorithms, Fall 2016

Group Assignment 2

Due: Tue, 10/25/16

Homework Policy:

1. Students should work on group assignments in groups of preferably three people. Each group submits to TEACH a zip file that includes their source code and their report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded, and the codes submitted to teach will be tested.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. You can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. More items might be added to this list. ☺

Let $P[1 \cdot\cdot m]$, and $Q[1 \cdot\cdot n]$ be two point sequences on the plane. Imagine two frogs Pfrog and Qfrog that are connected to each other with a leash would like to traverse these sequences together. In the beginning Pfrog is at $P[1]$, and Qfrog is at $Q[1]$. At each step, where the Pfrog is at $P[i]$ and Qfrog is at $Q[j]$, they can proceed in three different ways:

(A) Pfrog jumps forward to $P[i+1]$, and Qfrog stays at $Q[j]$,

(B) Qfrog jumps forward to $Q[j+1]$, and Pfrog stays at $P[i]$, or

(C) Pfrog and Qfrog jump forward together to $P[i+1]$ and $Q[j+1]$, respectively.

A leash is *useful* if Pfrog and Qfrog can use it to traverse P and Q, otherwise, it is too short. You want to buy a useful leash for the frogs. To save money you look for the shortest useful leash in the store. The input to your algorithm is P, Q, and a list available leashes $L = \{\ell_1, \ldots, \ell_t\}$. Your algorithm must find the shortest useful leash in L.

Report (70%). In your report, include the following items.

(1) Suppose P, Q and a real number ℓ are given. Let $J(P[1 \cdot\cdot m], Q[1 \cdot\cdot n], \ell)$ be *true* if and only if the frogs can traverse P and Q with a leash of length ℓ. Come up with a recursive relation for $J(P[1 \cdot\cdot m], Q[1 \cdot\cdot n], \ell)$. Argue that your recursive relation is correct.

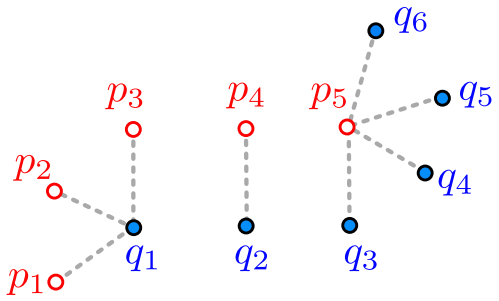

Figure 1: A valid sequence of moves for PFrog and Qfrog is $(P[1], Q[1]) \to (P[2], Q[1]) \to (P[3], Q[1]) \to (P[4], Q[2]) \to (P[5], Q[3]) \to (P[5], Q[4]) \to (P[5], Q[5]) \to (P[5], Q[6])$.

(2) Explain why the algorithm of part (1) is slow, and how can you speed it up (substantially) using Memoization.

(3) Change your memoized algorithm of part (2) into an iterative dynamic programming algorithm, and analyze its running time.

(4) Now, suppose that you are given a list of lengths $L = \{\ell_1, \ldots, \ell_t\}$, instead of just one number ℓ. Describe an efficient algorithm to pick the shortest useful leash from L. (Hint: use the algorithm of part (3) as a black box)

Code(30%) Write a program to solve the problem above. Your program reads from the input file "input.txt". Line 1 of the input is a single integer $1 \leq m \leq 1000$. Line 2 specifies P. It is a list of m points (pair of integers) separated by commas. Line 3 of the input is a single integer $1 \leq n \leq 1000$. Line 4 specifies Q. It is a list of n points (pair of integers) separated by commas. Line 5 of the input is a single integer $1 \leq t \leq 1000$. Line 6 specifies L. It is a list of t integers separated by commas.

Your output must be written in "output.txt". The output is a single number, the length of the shortest useful leash in L.

Sample Input (1):
2
(0,0),(2,0)
2
(0,1),(2,1)
6
5,0,1,3,12,5

Sample Output (1):
1

Sample Input (2):
2
(0,0),(2,0)
3
(0,1),(1,1),(2,1)
6
5,0,1,3,12,5

Sample Output (2):
3

Sample Input (3):
6
(0,0),(2,0),(3,0),(4,1),(3,2),(2,2)
5
(0,-1),(-1,0),(0,1),(2,1),(3,1)
8
5,0,1,3,2,5,12,18

Sample Output (3):
2

Submission Each group submits to TEACH a zip file that includes their source code (*which must be just one file with name "jump.cpp", "jump.java", or "jump.py"*) and their report. This file can be submitted by any member of the group, but all names must be listed in the submitted report. Each group, also, hands in a printed hard copy of the report in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded, and the codes submitted to teach will be tested.

Test your code with the sample test files (`http://web.engr.oregonstate.edu/~nayyeria/CS325/Fall16/hws/test2.zip`) before submitting them, to make sure there is no formatting error.

Do *not* submit a solution for the following problem, it is for practice.

Practice Problem. [1] Suppose you are given a sequence of integers separated by + and − signs; for example:

$$1 + 3 - 2 - 5 + 1 - 6 + 7.$$

You can change the value of this expression by adding parentheses in different places. For example:

$$\begin{aligned} 1 + 3 - 2 - 5 + 1 - 6 + 7 &= -1 \\ (1 + 3 - (2 - 5)) + (1 - 6) + 7 &= 9 \\ (1 + (3 - 2)) - (5 + 1) - (6 + 7) &= -17 \end{aligned}$$

Describe and analyze an algorithm to compute, given a list of integers separated by + and − signs, the maximum possible value the expression can take by adding parentheses. You may only use parentheses to group additions and subtractions; in particular, you are not allowed to create implicit multiplication as in $1 + 3(-2)(-5) + 1 - 6 + 7 = 33$.

[1] This problem is from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on dynamic programming is recommended.