CS325: Analysis of Algorithms, Winter 2020

Group Assignment 2*

Due: Tue, 2/4/20

Homework Policy:

1. Students should work on group assignments in groups of at most and preferably three people. Each group submits to TEACH a zip file that includes their source code and their *typeset* 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 rope would like to traverse these sequences together. In the beginning Pfrog is at $P[1]$, and Qfrog is at $Q[1]$. At each step, if 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 piece of rope 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 piece of rope for the frogs. To save money you look for the shortest useful piece in the store. The input to your algorithm is P, Q, and a list available ropes by their lengths $L = \{\ell_1, \ldots, \ell_t\}$. Your algorithm must find the shortest useful leash in L.

*The problem is from Jeff Erickson's book. Looking into similar problems from his lecture notes on recursion is recommended.

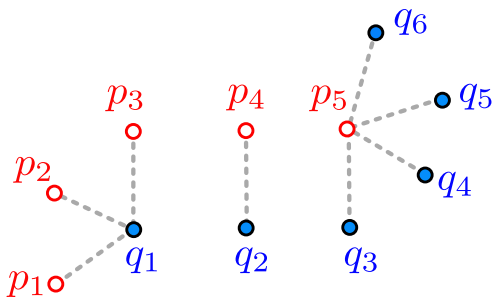

Figure 1: A valid sequence of moves for PFrog and Qfrog is $(P[1], Q[1]) \to (P[2], Q[1]) \to (P[3], Q[1]) \to (P[4], Q[2]) \to (P[5], Q[3]) \to (P[5], Q[4]) \to (P[5], Q[5]) \to (P[5], Q[6])$.

Report (60%). In your report, include the description of your algorithms, running time analysis, and proof of correctness. 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.

Code (40%). Submit a python program for part (a) of the problem. Your program will be tested against several test cases, for correctness and efficiency. For each test case, the program will be automatically stopped after 20 seconds if it is not done in that time. In this case, the group will miss the points of that test case. **Note:** it is important that your output is formatted as described below, since your codes will be tested automatically.

Specifically, you must implement the function "shortest_useful_rope" in the following code. The code you submit will be an implementation of this procedure in a file named "assignment2.py".

```
'''
    This file contains the template for Assignment1.  For testing it, I will place it
    in a different directory, call the function <count_crossing>, and check its output.
    So, you can add/remove  whatever you want to/from this file.  But, don't change the name
    of the file or the name/signature of the following function.

    Also, I will use <python3> to run this code.
'''

def shortest_useful_rope(input_file_path, output_file_path):
    '''
        This function will contain your code.  It wil read from the file <input_file_path>,
        and will write its output to the file <output_file_path>.
    '''
    pass

'''
    To test your function, you can uncomment the following command with the the input/output
    files paths that you want to read from/write to.
'''
# count_crossings('', '')

```

Extra Credit – Test cases (∼10%). Submit at most five test cases according to the format described below. We run all submitted programs against your test cases and you receive points proportional to the number of test cases that fail.

Input/Output The input file is composed of three lines. The first line specifies P. It is a list of m points (pair of integers) separated by commas. The second line specifies Q. It is a list of n

points (pair of integers) separated by commas. The third line specifies L. It is a list of t integers separated by commas. It is guaranteed that $1 \leq m, n, t, \leq 1000$.

The output file must composed of one line that contains a single integer that is the length of the shortest useful rope in L.

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

Sample Output (1):
1

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

Sample Output (2):
3

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

Sample Output (3):
2