CS515: Algorithms and Data Structures, Fall 2016

Homework 2*

Due: Tue, 10/25/16

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
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. Of course, 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. ☺

Problem 1. [35 pts] Let $P = (p_1, p_2, \ldots, p_n)$, and $Q = \{q_1, q_2, \ldots, q_n\}$ be two sequences of points on the plane specifying locations of stones. Imagine two frogs Pfrog and Qfrog that are connected to each other with a leash would like to hop through these sequences of stones 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 hops forward to p_{i+1}, and Qfrog stays at q_j.

(B) Qfrog hops forward to q_{j+1}, and Pfrog stays at p_i.

(C) Pfrog and Qfrog hop forward together to p_{i+1} and q_{j+1}, respectively.

Design and analyze an algorithm to compute the minimum length of a leash that allows Pfrog and Qfrog to perform a sequence of hops to reach p_n and q_n, respectively, while they are connected with the leash. For full credit your algorithms must accomplish the task in polynomial time.

---

*Some of the problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on recursion and dynamic programming is recommended.

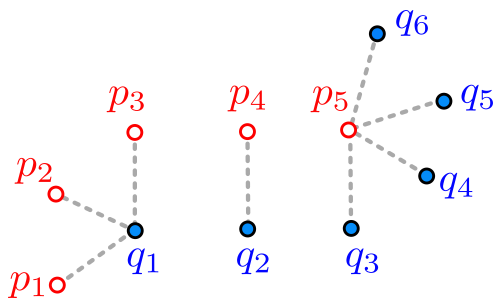


Figure 1: A valid sequence of moves for PFrog and Qfrog is $(p_1, q_1) \to (p_2, q_1) \to (p_3, q_1) \to (p_4, q_2) \to (p_5, q_3) \to (p_5, q_4) \to (p_5, q_5) \to (p_5, q_6)$.

Problem 2. **[35 pts]** Suppose we need to distribute a message to all the nodes in a rooted tree. Initially, only the root node knows the message. In a single round, any node that knows the message can forward it to at most one of its children. Design and analyze an algorithm to compute the minimum number of rounds required for the message to be delivered to all nodes. For full credit your algorithms must accomplish the task in polynomial time.

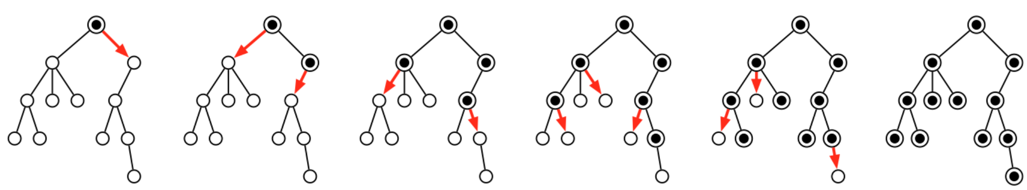
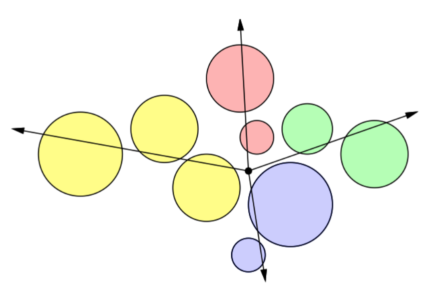


Figure 2: A message being distributed through a tree in five rounds.

Problem 3. Given a set C of n circles in the plane, each specified by its radius and the (x, y) coordinates of its center, compute the minimum number of rays from the origin that intersect every circle in C. Your goal is to design and analyze an efficient algorithm for this problem. For full credit your algorithms must accomplish the task in polynomial time.

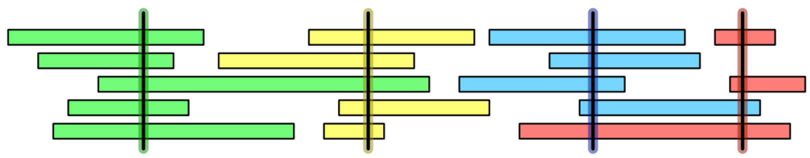


(a) **[20 pts]** Suppose it is possible to shoot a ray that does not intersect any balloons. Describe and analyze a greedy algorithm that solves the problem in this special case.

(b) **[10 pts]** Describe and analyze a greedy algorithm whose output is within 1 of optimal. That is, if m is the minimum number of rays required to hit every balloon, then your greedy algorithm must output either m or $m + 1$.

Here is a set of practice problems on asymptotic running time analysis, and recursion. Do *not* submit solutions for the following problems, they are just for practice.

Practice Problem A. Let X be a set of n intervals on the real line. We say that a set P of points stabs X if every interval in X contains at least one point in P. Describe and analyze an efficient algorithm to compute the smallest set of points that stabs X. Assume that your input consists of two arrays $X_L = [1..n]$ and $X_R = [1..n]$, representing the left and right endpoints of the intervals in X. Prove that your algorithm is correct.



A set of intervals stabbed by four points (shown here as vertical segments)

Practice Problem B. You are driving a bus along a highway, full of rowdy, hyper, thirsty students and a soda fountain machine. Each minute that a student is on your bus, that student drinks one ounce of soda. Your goal is to drop the students off quickly, so that the total amount of soda consumed by all students is as small as possible.

You know how many students will get off of the bus at each exit. Your bus begins somewhere along the highway (probably not at either end) and move s at a constant speed of 3.14 miles per hour. You must drive the bus along the highway; however, you may drive forward to one exit then backward to an exit in the opposite direction, switching as often as you like. (You can stop the bus, drop off students, and turn around instantaneously.)

Describe an efficient algorithm to drop the students off so that they drink as little soda as possible. Your input consists of the bus route (a list of the exits, together with the travel time between successive exits), the number of students you will drop off at each exit, and the current location of your bus (which you may assume is an exit). Prove your algorithm is correct and analyze its running time.