CSE Ph.D. Qualifying Exam, Fall 2019
Algorithms

Instructions:

Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total.

Questions:

1. **Greedy**

   You are given n distinct points and one line l on the plane and some constant $r > 0$. Each of the n points is within distance at most r of line l (as measured along the perpendicular). You are to place disks of radius r centered along line l such that every one of the n points lies within at least one disk. Devise a greedy algorithm that runs in $O(n \log n)$ time and uses a minimum number of disks to cover all n points; prove its optimality.

2. **Greedy and DP algorithms**

   Consider the following game. You are given a sequence of n positive numbers $(a_1, a_2, ..., a_n)$. Initially, they are all colored black. At each move, you choose a black number a_k and color it and its immediate neighbors (if any) red (the immediate neighbors are the elements a_{k-1}, a_{k+1}). You get a_k points for this move. The game ends when all numbers are colored red. The goal is to get as many points as possible.

   (a) Describe a greedy algorithm for this problem. Verify that it does not always maximize the number of points and give a tight approximation ratio (i.e., provide a family of instances where the greedy algorithm returns solutions that reach this bound, and an informal proof that, on any instance, the solution returned by the greedy algorithm will not exceed that bound).

   (b) Describe and analyze an efficient dynamic programming algorithm for this problem that returns optimal solutions. (Linear time is possible.)

3. **Dynamic programming**

   You are given a sorted set of points $P = (P_1, P_2, ..., P_n)$ on a line. Given a constant k, show how to select a subset of $k-1$ of these points, say (still in sorted order) $(P_{j_1}, ..., P_{j_{(k-1)}})$, so as to partition the segment from P_1 to P_n into k pieces that are as close to equal in length as possible. Specifically, writing $L = (P_n - P_1)/k$, we want to minimize the square error

$$(P_{j_1} - P_1 - L)^2 + \sum_{i=1}^{k-2} (P_{j_{i+1}} - P_{j_i} - L)^2 + (P_n - P_{j_{k-1}} - L)^2$$

   Describe and analyze an algorithm for this problem that runs in $\Theta(kn^2)$ time.

4. **NP: Dinner with Frenemies**

   Prove that the following decision problem is NP-complete. Given n students and a set of pairs of students who are enemies, is it possible to arrange a dinner around a round table so that two enemies do not sit side by side? Remember to include all the steps of the NP-completeness proof.