CS 3511: Algorithms Honors, Homework 3

Instructor: Prof. Merrick Furst
TAs: Karthik Gopalakrishnan, Pranathi Reddy Tupakula, Jiachen Li

Please submit a LaTeX-formatted PDF containing your solutions on T-Square, by Mar 2, 3:00 pm.

1 Height Inversions

N people are standing in a queue to buy tickets to watch a movie. Your friend is one of them, having just entered the queue to buy two tickets while you wait for him at the waiting area. While queues work in a first-in-first-out (FIFO) fashion and most people (including you) think they are fair, you're lamenting at how the line is so visually unappealing because the people aren't ordered in increasing order by height. So to kill time, you decide to sit and count the number of pairs of people in the queue such that the person ahead in the queue is taller than the person behind.

(10 points) Specifically, given an array A containing N heights (in centimeters), devise an $O(N \log N)$ divide-and-conquer algorithm to count the number of pairs (i, j) such that $A[i] > A[j]$ for $i < j$ and analyze its run-time complexity.

2 Extrema of Discrete Functions

We're working with discrete functions $f : D \to \mathbb{Z}$, where D is an **ordered** domain set (on which conditions are imposed in the questions below) and $\mathbb{Z}$ is the set of all integers. Suppose you know that in the ordering specified by D, f decreases up to a point $x \in D$ and then increases, i.e., f attains a minima at x in D.

(10 points) If D is finite, give an $O(\log |D|)$ algorithm to find the minima of f.

(10 points) Suppose D is any countably infinite set (such as $\mathbb{N}$, the set of natural numbers) and x is finite but quite possibly a very large value. Assume the existence of an order-preserving bijection between D and $\mathbb{N}$. Devise an efficient (faster than a linear scan over D) algorithm to find the minima of f, argue that it is correct and analyze its run-time complexity.

(no points, just a fun observation) Some of you might notice that if the constraints are relaxed further to have $f : D \to \mathbb{R}$ and D as an uncountably infinite convex set (for example, $D = \mathbb{R}$, the set of real numbers), the problem becomes a convex optimization problem. Additionally, if f doesn't necessarily follow the decrease-increase observation made above, the problem becomes the more general non-convex optimization problem, which underlies most of modern-day machine learning and for which no provably correct and efficient algorithm is known.

3 Fun with Maps!

You're in the infamous game show "Fun with Maps!", hosted by Dr. Sheldon Cooper. Sheldon gives you a map of a city, comprising of buildings and roads connecting them, and he tells you that one of the buildings (let's call it X) contains a hidden treasure. You can only find the treasure by choosing a building B from the map and asking Sheldon if the treasure is present in it, but since Sheldon is rude and has a history of offending his contestants by calling their choices stupid, you want to minimize how often you ask him this question. That apart, he also gives you game-points inversely proportional to the total number of times you asked him the question to find the treasure.

Specifically, this is what Sheldon does when you ask him to **verify** if the treasure is in a building B:

- If $B == X$: return X
- Else: return a building B' adjacent to B that lies on the path from B to X

(10 points) Sheldon gives you a map in which n buildings are located in a line, with roads connecting every consecutive pair of buildings (i.e., a n-vertex path P_n). Come up with an algorithm to find the treasure vertex X by asking Sheldon for verification not more than $O(\log n)$ times and formally prove that it is correct.

(10 points) Sheldon now gives you a map corresponding to a (general) tree. Recall that a tree with n vertices has $n - 1$ edges and that there is exactly one path between any two vertices in a tree. Come up with an algorithm to find the treasure vertex X by asking Sheldon for verification not more than $O(\log n)$ times and formally prove that it is correct.