Homework 8

Due: November 23, 2021

This homework must be typed in LaTeX and handed in via Gradescope.

Please ensure that your solutions are complete, concise, and communicated clearly. Use full sentences and plan your presentation before you write. Except in the rare cases where it is indicated otherwise, consider every problem as asking you to prove your result.

Problem 1

1. Suppose we are given a sequence $Q = \{q_0, q_1, \ldots, q_{n-1}\}$ of points in $\mathbb{R}^2$ such that q_0 is the anchor and $q_1, q_2, \ldots, q_{n-1}$ are the other vertices on the convex hull in counterclockwise order starting from the anchor. Describe an algorithm that given Q and a new point $(x, y) \in \mathbb{R}^2$ construct in linear time a sequence Q' which is the convex hull of the points in Q and the new point (x, y). Show that your proposed algorithm is correct and analyze its running time.

2. In the *on-line convex-hull problem*, we are given the set P of n points one point at a time. After receiving each point, we are to compute the convex hull of the points seen so far

   (a) Propose an algorithm that uses the Graham scan so that after n points the overall running time is $O(n^2 \log n)$. Prove the correctness and analyze the running time of the proposed algorithm.

   (b) Show how to improve this slightly, by showing we can solve the online convex hull problem in $O(n^2)$. Prove the correctness and analyze the running time of the proposed algorithm.

Problem 2

Line segments and polygons are used to model geometric objects in computer graphics, video games, and computer-aided design. Suppose you are given a set S, of n line segments, in no particular order. Describe an efficient algorithm for determining whether the segments of S form a polygon, P, and if so, give a polygonal representation for P. You may allow P to be non-simple (that is, self-intersecting), but P must be a single cyclical chain of vertices connected by the *unique* segments in S. An easy way to think of this definition a polygon P is that P must be able to be formed by one piece of string, where the beginning of the string touches the end of the string, and no segments of the string lie on top of other segments. What is the running time of your algorithm?

Problem 3

Show how to extend the Rabin-Karp method to handle the problem of looking for a given $m \times m$ pattern in an $n \times n$ array of characters. Describe the overall algorithm and how you would approach computing the hashcodes used in Rabin-Karp. Your algorithm should not use more than $O(m)$ additional memory and should require at most $O(n^2 m)$ time. Prove the correctness of your solution, analyze its running time and memory space utilization.