Written Problems

Don’t forget to prove that your solutions meet the relevant time and/or space bounds!

1. Exercise 20.2–7 in CLRS.

2. You find yourself standing in a huge maze with high featureless walls. You have a large number of brightly colored coins. Give an algorithm for escaping that uses time and pennies linear in the size of the maze. (My understanding is that this is one of the oldest algorithms known, over 2,500 years old.)

3. Given a sequence of \( n \) values \( x_1, x_2, \ldots, x_n \), you will need to answer queries of the form: given \( i, j \) (with \( i \leq j \)), what is the smallest value in the subsequence \( x_i, \ldots, x_j \)? Show how to answer queries in \( O(\lg n) \) time per query using at most \( O(n \lg n) \) space.

4. Give an \( O(|V| + |E|) \) time algorithm that, given a graph, determines whether or not it is a forest.

5. You observe the price of a stock for the last \( n \) days and wonder wistfully how much money you could have made by trading in it. Give an algorithm that finds the best day \( i \) to buy and day \( j > i \) to sell to maximize your profit. For 2/3 credit, your algorithm can run in \( O(n \lg n) \) time; for full credit, it must be \( O(n) \) (so that you can afford to run it for all the stocks on the NASDAQ).

6. (Those in 858 only) Let \( S = \{1, 2, \ldots, n\} \) and \( f : S \to S \). For \( R \subseteq S \), define \( g(R) = \{ f(x) | x \in R \} \). Design an \( O(n) \) time algorithm for determining the largest \( R \subseteq S \) such that \( g(R) = R \). (If you aren’t familiar with function notation, see Appendix B.3 in CLRS. If you are still confused, you are always welcome to see the course staff at recitation or office hours.)

7. What suggestions do you have for improving this assignment in the future?

Submission

Submit your work electronically as usual.