Exam #2 - Data Structures and Algorithms
COMP 251-252 B - March 15, 2007

1. Range Searching

Assume you have a data-base consisting of N items, each of which is represented by two coordinates X and Y. Therefore the data may be viewed as a set of N points in the plane. We want to be able to answer range queries efficiently, that is, how many points lie in a given query rectangle with sides parallel to the coordinate axes.

(a) Design an algorithm based on the *locus* method that constructs a data structure in $O(N^2)$ time, that uses $O(N^2)$ space, and that allows queries to be answered in O(log N) time.

(b) Prove the correctness of your algorithm.

(c) Prove the three complexities of your algorithm.

2. The Largest Gap Problem

You are given a set *S* of *n* distinct, but otherwise arbitrary, real numbers on a real Random Access Machine with computing power augmented to include the computation of *ceiling* and *floor* functions in O(1) time per operation. The *largest gap* is the largest interval between any two *consecutive* numbers in *S*.

(a) Describe in as much detail as you can an algorithm for computing the largest gap in O(*n*) worst-case time.

(b) Prove the correctness of your algorithm.

(c) Prove the complexity of your algorithm.

3. Dynamic Programming

(a) Describe the *dynamic programming* algorithm for computing the *edit-distance* between two sequences of letters. Apply it to the problem of converting the sequence ***must*** to the sequence ***mater***. Use a cost of ***one*** for *insertions* and for *deletions*. For *substitutions* use a cost of ***one*** when the elements are *distinct* and ***zero*** when the elements are *identical*. Show the complete table used by the algorithm and highlight the minimum-cost path on this table.

(b) What is its worst-case complexity of your algorithm? Explain.

(c) How did you obtain the minimum-cost path?

This question is to be answered only by COMP 252 students

---

1. Algorithms on Sequences

You are given a set S containing N real numbers, and an additional number X.

(a) Design an O(N log N) worst-case time algorithm to determine if there are two elements of S whose sum is exactly X.

(b) Prove the correctness of your algorithm.

(c) Prove your algorithm runs in O(N log N) time.