

Homework 7, Due Monday, February 26, 2013

Problem 1 (10 points):

Chapter 6, Page 323, Problem 11.

Problem 2 (10 points):

Chapter 6, Page 327, Problem 16

Problem 3 (10 points):

Chapter 6, Page 329, Problem 20.

Problem 4 (10 points):

Chapter 6, Page 334, Problem 28.

Problem 5 (30 points):

The longest common subsequence problem is given two sequences, A and B , find the longest sequence that is a subsequence of both of them. This can be done by an $O(nm)$ dynamic programming algorithm. The basic algorithm requires $O(nm)$ space to find the the subsequence. This can be prohibitive if n and m are large (note that space is a bigger constraint than time). The text describes a $O(n+m)$ space algorithm for this problem. Implement the algorithm from the text. (Note that the text describes the problem as string alignment, and makes it slightly more general - LCS is essentially the same problem, just slightly cleaner.)

Implement the $O(nm)$ time, $O(n+m)$ space algorithm that finds the longest common subsequence of A and B .

For test data use random sequences over $\{a, c, t, g\}$. Report the length of the Longest Common Sequences for a range of values. Run for as large an n as you can so that the run time is no more than 5 minutes.

Turn in your code and the results. Grading will be based on a successful implementation of the space efficient algorithm.