

1. Written assignment: Printed textbook 5.24 and 6.1.
2. Coding assignment: Consider the longest increasing subsequence problem: Given an array A of integers, find the length of the longest subsequence of A such that the elements in the subsequence are in an increasing order. For example, if $A = \{8, 4, 12, 2, 10, 3, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15\}$. Then the length of the longest increase subsequence is 6. Do the following:
 - (a) Implement an $O(n^2)$ time dynamic programming algorithm for the longest increasing subsequence problem.
 - (b) Implement an $O(n \log n)$ time dynamic programming algorithm for the longest increasing subsequence problem, using the method we discussed in class.
 - (c) Write a subroutine to generate sequences of 10k, 20k, ..., 100k random numbers and use them as the inputs to test the above two algorithms. Compare the results of both algorithms and make sure that they match. Plot the running times of both algorithms as functions of the input sizes.