

CSC282 Fall 2005 Homework #7

1. In the sequence of matrix multiplication problem we encountered the fact that there are lots of subsequences. Consider the sequence $0, 1, \dots, n$.

A. Show that there are 2^{n+1} distinct subsequences, including the empty one.

B. Show that there are fewer than $n^3/4$ subsequences of length three or less.

C. Conclude that there are at least 2^n distinct subsequences of length four or more, for $n \geq 6$ (or maybe 7...).

2. Read both parts before you begin.

A. Let E be an array of n distinct integers. Give an algorithm to find the length of the longest increasing subsequence of entries in E . The subsequence is not required to be contiguous in the original sequence. *E.g.* for $[11, 17, 5, 8, 6, 4, 7, 12, 3]$ the answer is $[5, 6, 7, 12]$. Analyze the worst-case running time and space requirements of your algorithm.

B. In part A the original sequence does not “wrap around”, so $[5, 8, 12, 17]$ is not a legal answer. But now suppose you can consider such solutions. What changes?

3. (double points for this one) The *partition problem* is: given a sequence of n nonnegative integers as input, to find a way to partition this sequence into two disjoint subsequences so that the sums of the integers in the two subsequences are equal. That is, with nonnegative integers s_1, s_2, \dots, s_n that sum to S , find a subset I of $1, 2, \dots, n$ such that

$$\sum_{i \in I} s_i = \sum_{j \notin I} s_j = S/2,$$

or determine that there is no such subset. Give a DP algorithm for the partition problem. Analyze the worst-case running time and space requirements of your algorithm as functions of n and S .

4. Suppose coin denominations in a country are $c_1 > c_2 > \dots > c_n$. The *coin changing problem* is: given a sequence of denominations and an amount, a cents, as input — to determine the minimum number of coins needed to make a cents in change. Assume that $c_n = 1$, so you can always get a solution.

A. Describe a greedy algorithm, and explain how it would work for \$1.43 in US coins.

B. Again with US coinage, prove your greedy algorithm is optimal, that is uses the minimal number of coins to make a cents.

C. In fact, prove your greedy algorithm is also optimal for any denominations that are powers of c (*i.e.* the denominations are c^0, c^1, \dots, c^k for some integers $c > 1$ and $k \geq 1$).

D. However, give a set of denominations for which your greedy algorithm is not optimal (still include the penny, so all a 's have a solution.)

E. Give a DP algorithm for this problem. Analyze the worst-case running time and space requirements of your algorithm as functions of n and a .

5. Give an algorithm to determine how many distinct ways there are to give a cents in change using coins from the denomination set $\{1, 5, 10, 25, 50\}$ cents. *E.g.* there are six ways to change 17 cents.

6. CLRS 15.2-3.

7. CLRS 15.3-2

8. CLRS 15.4-3