Homework 4: Dynamic Programming

1 The mixture sequence problem

A sequence of n natural numbers $C = \langle c_1, c_2, \ldots, c_n \rangle$ is said to be a ***mixture*** of a sequence of n natural numbers $A = \langle a_1, a_2, \ldots, a_n \rangle$ and another sequence of n natural numbers $B = \langle b_1, b_2, \ldots, b_n \rangle$ if and only if for every $1 \leq i \leq n$ we have $c_i = a_i$ or $c_i = b_i$. In other words, for i from 1 to n, we pick either a_i or b_i and make it c_i. We say $\sum_{1 \leq i \leq n} c_i$ is a ***mixture sum*** of A and B.

The mixture sequence problem is defined as follows:

- Given two sequences of n natural numbers of $A = \langle a_1, a_2, \ldots, a_n \rangle$ and $B = \langle b_1, b_2, \ldots, b_n \rangle$,
- and given a natural number m,
- determine whether there exists a mixture C of A and B such that the the mixture sum of C equals m, and
- if such a mixture sequence does exist print out the mixture sequence; otherwise print out a message saying there is no such mixture sequence.

2 A Dynamic-Programming Approach

The following is a dynamic programming approach to solve the problem:

- For $0 \leq l \leq n$ and $0 \leq s \leq m$, let $hasSolution(l, s)$ be a boolean value indicating whether there is a solution to the instance of mixture sequence problem where $A = \langle a_1, a_2, \ldots, a_l \rangle$, $B = \langle b_1, b_2, \ldots, b_l \rangle$, and the target sum is s.
- First, in the case of $l = 0$, $hasSolution(0, s)$ is true if and only if $s = 0$; otherwise $hasSolution(0, s)$ is false.
- Second, in the case of $0 < l \leq n$, $hasSolution(l, s)$ is true if and only if either $hasSolution(l-1, s-a_l)$ is true (in this situation, a_l can be selected to generate a solution mixture sequence) or $hasSolution(l-1, s-b_l)$ (in this situation, b_l can be selected to generate a solution mixture sequence) is true (or both are true); otherwise $hasSolution(l, s)$ is false.
- You can use a two-dimensional array to keep tract of the values of $hasSolution(l, s)$ for every $0 \leq l \leq n$ and every $0 \leq s \leq m$. The key thing of the dynamic programming approach is to determine $hasSolution(l, s)$ for every $0 \leq l \leq n$ and every $0 \leq s \leq m$ only exactly once. This can be done either recursively with the checking of the memory to see whether $hasSolution(l, s)$ has been determined before we actually recurse down to solve the problem, or it can

be done using a two-level nested loop with the outer loop iterates l from 0 to n and the inner loop iterates s from 0 to m.

- After you have determined whether $hasSolution(l, s)$ is true or not for every $0 \leq l \leq n$ and every $0 \leq s \leq m$ systematically based on the descriptions above, if $hasSolution(n, m)$ is true, there is a solution to the original problem instance. In that situation, you can find a solution mixture sequence by starting from the entry for $hasSolution(n, m)$ in the two-dimensional array and then tracing down throughout other relevant $hasSolution(l, s)$'s stored in the two-dimensional array.

3 The Programming Assignment

Implement the divide-and-conquer approach above or some other divide-and-conquer approach of your own into a program. Your program should interact with the user like what you did in Homework 1. Check and report how much time your programs for Homework 1 and Homework 1 and your new program for this homework will take respectively over the problem instances posted on our class web site. Report the solutions found by your program.