Programming 1C: A branch-and-bound approach to prune the search space

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 Branch-and-Bound Approach

The following is a possible branch-an-bound approach to solve the problem:

- We refer to an empty sequence $C' = \langle \rangle$ of no elements as a partial mixture sequence of length 0. For $1 \leq l < n$, let's refer to a sequence $C' = \langle c_1, c_2, \ldots, c_l \rangle$ as a partial mixture sequence (of A and B) of length l if and only if $c_i = a_i$ or $c_i = b_i$ for $1 \leq i \leq l$. We use the notation $PartialSum(C')$ to refer to the sum of all the elements in the partial mixture sequence C'.
- For $0 \leq l \leq n$, let's define $L_i = \sum_{l+1 \leq k \leq n} \min(a_k, b_k)$ and $U_l = \sum_{l+1 \leq k \leq n} \max(a_k, b_k)$. In other words, L_l is the minimum (lower bound) of the sum of the $l+1$ th element to the n th (the last) element in any mixture sequence C derived from A and B while U_l is the maximum (upper bound) of the sum of the $l+1$ th element to the nth (the last) element in any mixture sequence C derived from A and B.
- Observation 1: We can search for a mixture sequence C with a target mixture sum m by starting from an empty partial mixture sequence C' of length $l = 0$ and then iteratively trying to extend the current partial mixture sequence $C' = \langle \ldots, c_l \rangle$ of length l one step further by appending either a_{l+1} or b_{l+1} to the end of the current partial mixture sequence C'. Naively, for a partial mixture sequence C' of length l there are two possible branches

to explore for further extensions into a final mixture sequence C with the target mixture sum m. However, Observation 2 below shows that at time we can completely cut off any search along these two branches completely by checking the bounds L_l and U_l against $PartialSum(C')$ and the target sum m.

- Observation 2: Note that whatever way we extend a partial mixture sequence $C' = \langle \ldots, c_l \rangle$ into a mixture sequence $C = \langle c_1, c_2, \ldots, c_n \rangle$, the resulting mixture sum must be at least $PartialSum(C') + L_l$ and at most $PartialSum(C') + U_l$. If the target sum m does not fall into the range of $[PartialSum(C') + L_l, PartialSum(C') + U_l]$, we know that it is impossible to extend the current partial mixture sequence C' into a mixture sequence ending in the target mixture sum of m. In this case, we can completely cut off any further search effort to extend the current partial mixture sequence C'.

3. **Implementation and experiments to do**:
Implement the branch-and-bound scheme above for solving the mixture sequence problem.
Note: You may consider using **this** **recursive exhaustive branching search** implementation as the basis and then incorporate the branch-and-bound scheme into it.

Experiment #A: Run your new program over at least the test cases 1A~1D in Programming #1B. **Please report whether your new program can get the same results correctly as your program for Programming #1B did.**

Experiment #B: Consider the following two sequences:

A =<18, 29, 31, 42, 55, 66, 71, 85, 91, 103, 114, 128, 135, 140, 155, 81, 93, 17, 24, 65, 39, 27, 58, 19, 130, 141, 182, 153, 104, 115, 18, 29, 31, 42, 55, 66, 71, 85, 91, 103, 114, 128, 135, 140, 155> and

B =<81, 93, 17, 24, 65, 39, 27, 58, 19, 130, 141, 182, 153, 104, 115, 18, 29, 31, 42, 55, 66, 71, 85, 91, 103, 114, 128, 135, 140, 155, 81, 93, 17, 24, 65, 39, 27, 58, 19, 130, 141, 182, 153, 104, 115 >.

Use your new program to search for a mixture sequence with the sum of
(i) 655, (ii) 3869, (iii) 3901, (iv) 4139, (v) 99991 respectively.
Report your findings on cases that the program can finish within 1 day. For each case, please report whether your new program can find a mixture sequence (and include the sequence if it does find one). How much time on average does it take to finish one case?

Experiment #C: Try to use your old program for Programming #1B to repeat what you are required to do in the Experiment B above. **Report your findings on cases that the program can finish within 1 day.**