CSC 226 Lab 8

Dynamic Programming – Coin In a Line Game Problem

In this game, which we will call the coins-in-a-line game, an even number, n, of coins, of various denominations from various countries, are placed in a line. Two players, who we will call Alice and Bob, take turns removing one of the coins from either end of the remaining line of coins. That is, when it is a player's turn, he or she removes the coin at the left or right end of the line of coins and adds that coin to his or her collection. The player who removes a set of coins with larger total value than the other player wins, where we assume that both Alice and Bob know the value of each coin.

Example

coins [] = { 6, 9, 1, 2, 16, 8}

trial 1:
coins [] = { 6, 9, 1, 2, 16, 8} **,** Alice picks 8
coins [] = { 6, 9, 1, 2, 16}**,** Bob picks 16
coins [] = { 6, 9, 1, 2}, Alice picks 6
coins [] = { 9, 1, 2}**,** Bob picks 9
coins [] = {1, 2}**,** Alice picks 2
coins [] = {1}**,** Bob picks 1
Alice: 8+6+2 =16 Bob: 16+9+1=26 => **Alice Lost**

So clearly picking up the best in each move is not good for Alice. What else Alice can do to win the game.

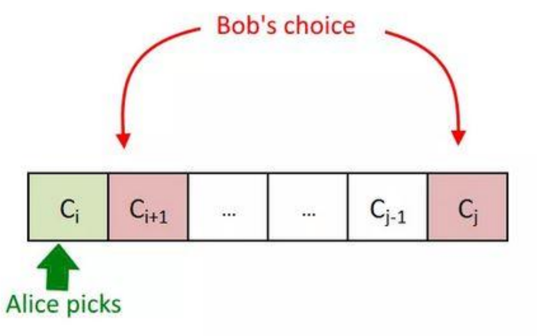
trial 2:
coins [] = { 6, 9, 1, 2, 16, 8} **,** Alice picks 6
coins [] = { 9, 1, 2, 16, 8}**,** Bob picks 9
coins [] = { 1, 2, 16, 8}, Alice picks 1
coins [] = 2, 16, 8}**,** Bob picks 8
coins [] = {2, 16}**,** Alice picks 16
coins [] = {2}**,** Bob picks 2
Alice: 6+1+16 =23 Bob: 9+8+2=19 => **Alice Won**

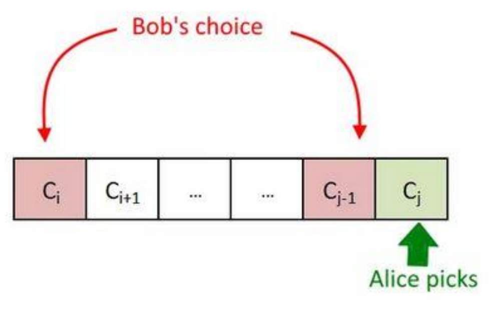
Solution

MV(i, j) = maximum value the Alice can collect from i'th coin to j'th coin.

Alice has 2 options:

1. Pick coin I (from starting)
   Maximum value that can be achieved:
   $V_i + Min\{MV(i+2,j), MV(i+1, j-1)\}$
2. Pick coin j (from ending)
   Maximum value that can be achieved:
   $Vj + Min\{MV(i+1,j-1), MV(i, j-2)\}$





So now we need to decide whether Alice should pick ith coin or jth coin. Alice will pick the coin which ever gives the more value considering 2 moves ahead.

MV(i, j) = Max $\{ V_i + Min\{MV(i+2,j), MV(i+1, j-1)\} , Vj + Min\{MV(i+1,j-1), MV(i, j-2)\}\}$

MV(i, j)	=Vi	Base Case i=j
	=Max(Vi, Vj)	Base Case j=i+1
	= **Max** $\{ V_i + Min\{MV(i+2,j), MV(i+1, j-1)\} , V_j + Min\{MV(i+1,j-1), MV(i, j-2)\}\}$	

Please use memoization if using top down(recursive) approach

Test Cases:

Coins[]= {8,15,3,7}	output=22
Coins[]={2,2,2,2}	output=4
Coins[]= { 20, 30, 2, 2, 2, 10}	output= 42