**Winning the coin game**

This task is done using only one function called best\_score( ) function. This function takes in two parameters which are called pile1 and pile2. The aim of this game is for player 1 to get the maximum value given player 1 starts first. The function starts by generating a memoization table of auxiliary space of O(x), where is the number of coins in pile 1 and is the number of coins in pile 2. The time complexity to make this table is also O(x). Because it has to go through every position in the grid.

The function then proceeds to generate the base case of this problem, which is when both piles are empty, pile1 is empty and pile2 is empty. These are the base cases because player 1 and 2 do not have to make decision when both piles are empty. They also don’t need to make decision when one pile is empty, since player 1 starts first he will take the alternating coin starting from the top of the pile. Same goes to player 2.

The memoization table is then filled up slowly based on the base cases. For example, when making decision at memo[j][i], player 1 will refer to both memo[j-1][i] and memo[j][i-1] to see which pile would yield a greater value if he chooses the coin from that pile. This process will yield a time complexity of O(x), where is the number of coins in pile 1 and is the number of coins in pile 2.

When it comes to backtracking, we just need to refer to the previously filled memoization table to track back coin from which pile did player 1 choose in every round. This process yields a time complexity of O(+), where is the number of coins in pile 1 and is the number of coins in pile 2.

The total complexity of this function is O(x).