

# Dynammic Programming

## Main Focus: Optimal Game Strategy

### Optimal Strategy for a Game:

Consider a row of  $n$  coins of values  $v_1, \dots, v_n$ , where  $n$  is even (since it's a two player game). We play this game with the opponent. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. **Determine the maximum possible amount of money we can definitely win if we move first.**

**We can reframe the question above in a easier way to understand for us as-**

Suppose we are given  $n$  pots, each with some number of gold coins, are arranged in a line. You are playing a game against another player. You take turns picking a pot of gold. You may pick a pot from either end of the line, remove the pot, and keep the gold pieces. The player with the most gold at the end wins. Develop a strategy for playing this game.

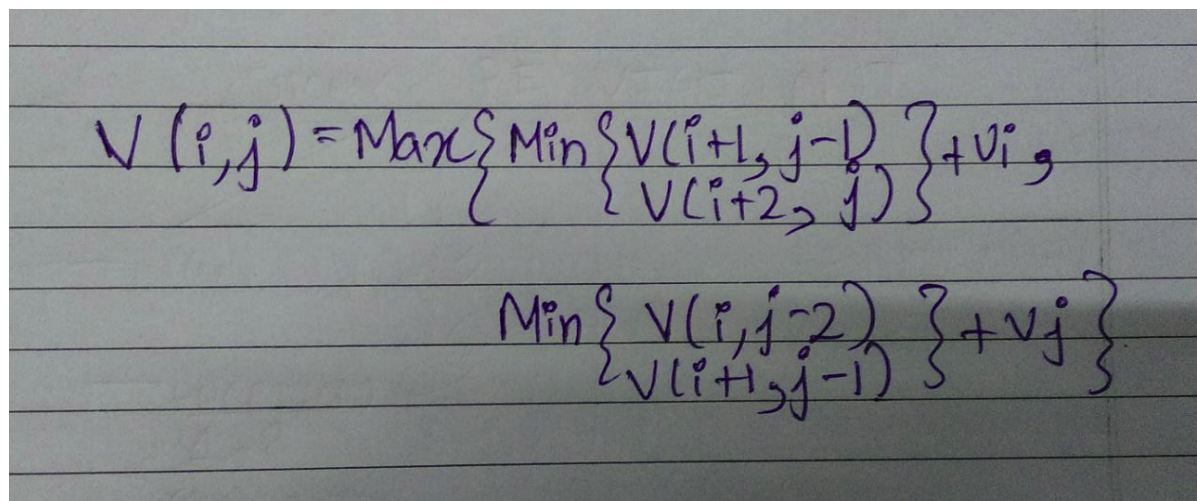
#### ###Approach:

Let us solve the problem using our DP technique. For each turn either we or our opponent selects the coin only from the ends of the row. Let us define the subproblems as:

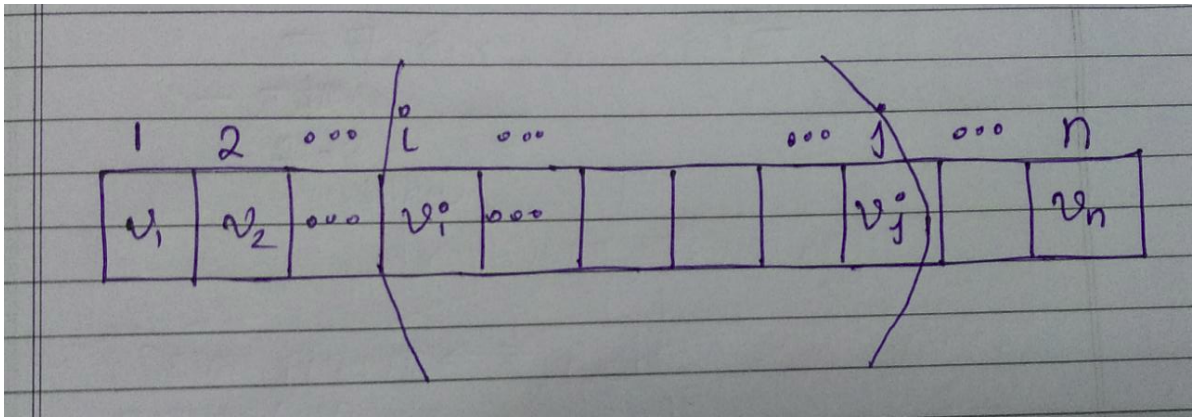
$V(i, j)$ : denotes the maximum possible value we can definitely win if it is our turn and the only coins remaining are  $v_i \dots v_j$ .

**Base Cases:**  $V(i, i)$ ,  $V(i, i + 1)$  for all values of  $i$ .

From these values, we can compute  $V(i, i + 2)$ ,  $V(i, i + 3)$  and so on. Now let us define  $V(i, j)$  for each sub problem as:

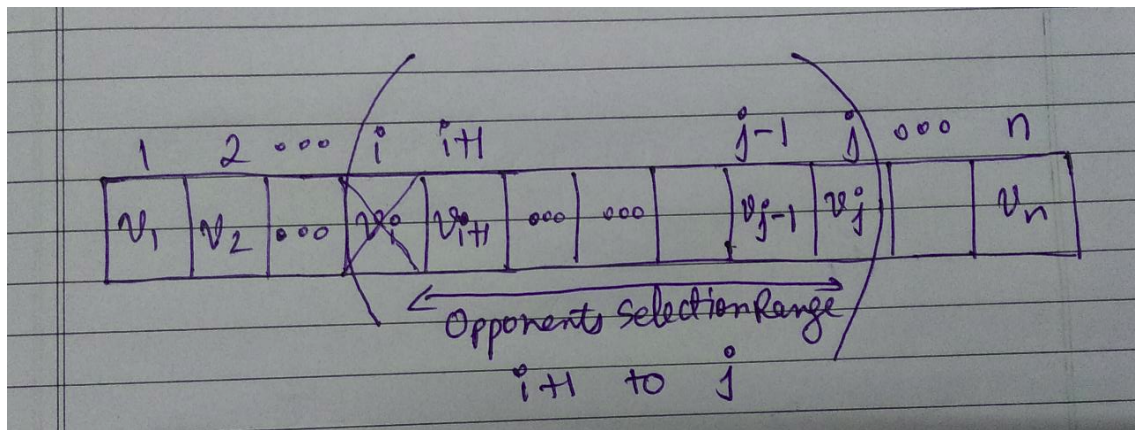

$$V(i, j) = \text{Max} \left\{ \text{Min} \left\{ \begin{array}{l} V(i+1, j-1) \\ V(i+2, j) \end{array} \right\} + v_i, \right. \\ \left. \text{Min} \left\{ \begin{array}{l} V(i, j-2) \\ V(i+1, j-1) \end{array} \right\} + v_j \right\}$$

In the recursive call, we have to focus on  $i^{\text{th}}$  coin to  $j^{\text{th}}$  coin ( $v_i \dots v_j$ ) coin. Since, it is our turn to pick the coin, we have two possibilities: either we can pick  $v_i$  or  $v_j$ .



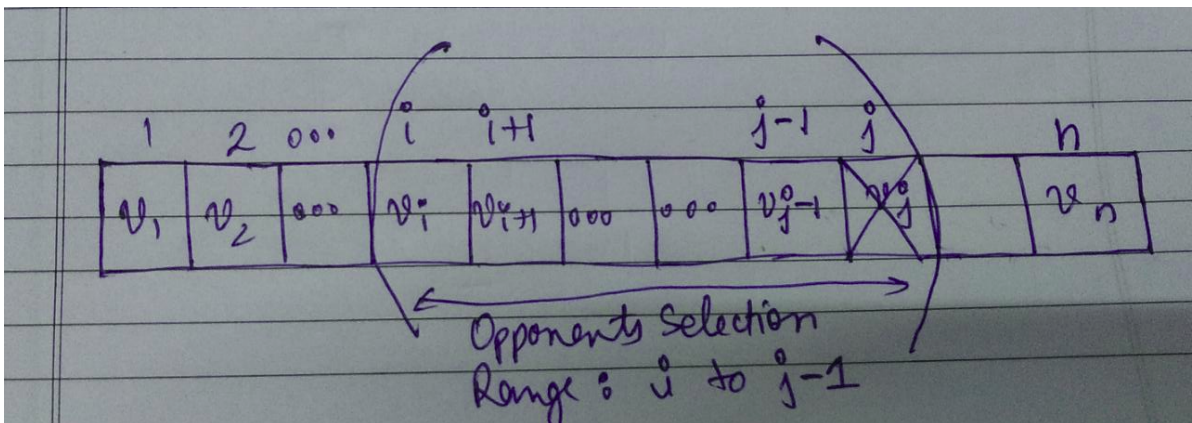
**NOTE:** In the figures given below, the Outer Max indicates that we have to select the coin which gives maximum value. Now, we focus and get:

1. **Selecting the  $i^{th}$  coin:** If we select the  $i^{th}$  coin, then the remaining range is from  $i + 1$  to  $j$ . Since, we selected the  $i^{th}$  coin then the remaining range is from  $i + 1$  to  $j$ . Since, we selected the  $i^{th}$  coin, we get the value  $v_i$  for that. From the remaining range  $i + 1$  to  $j$ , the opponents can select either  $(i + 1)^{th}$  coin or  $j^{th}$  coin. But, the opponents selection should be minimized as much as possible. It can be seen in the below figure:



2. **Selecting the  $j^{th}$  coin:**

It is literally the same argument as above. If we select the  $j^{th}$  coin, then the remaining range is from  $i$  to  $j - 1$ . Since we selected the  $j^{th}$  coin we get the value  $v_j$  for that. From the remaining range  $i$  to  $j - 1$ , the opponent can select either the  $i^{th}$  coin or the  $(j - 1)^{th}$  coin. But the opponent's selection should be minimized as much as possible. The figure below depicts the same:



**Subproblems:**  $i$  can range from 1 to  $n$  and  $j$  can range from 1 to  $n$ . There are a total of  $n^2$  subproblems and each takes  $O(1)$  and the total time complexity is  $O(n^2)$ .

