

Agenda

- Problem: Edit Distance
- Problem: Unique Paths In A Grid
- Problem: Max Sum Path in Binary Tree
- Problem: Optimal Strategy For A Game

Edit Distance

Given two strings **A** and **B**, find the minimum number of steps required to convert **A** to **B**. (each operation is counted as 1 step.)

You have the following 3 operations permitted on a word:

- Insert a character
- Delete a character
- Replace a character

Examples:

Input 1:

A = "abad"

B = "abac"

Output 1:

1

Explanation 1:

Operation 1: Replace d with c.

Input 2:

A = "Anshuman"

B = "Antihuman"

Output 2:

2

Explanation 2:

=> Operation 1: Replace s with t.

=> Operation 2: Insert i.

Unique Paths In A Grid

Given a grid of size $n * m$, let's assume you are starting at $(0, 0)$ and your goal is to reach $(n-1, m-1)$. At any instance, if you are on (x, y) , you can either go to $(x, y + 1)$ or $(x + 1, y)$.

Now consider if some obstacles are added to the grids. How many unique paths would there be?

An obstacle and empty space are marked as 1 and 0 respectively in the grid.

Examples:

Input:

```
[  
  [0,0,0],  
  [0,1,0],  
  [0,0,0]  
]
```

Output: 2

Input:

```
[  
  [0,1,0,1],  
  [0,0,0,0],  
  [0,0,0,0],  
  [0,1,1,0]  
]
```

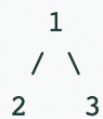
Output: 4

Max Sum Path in Binary Tree

Given a binary tree, find the maximum path sum. The path may start and end at any node in the tree.

Example :

Input 1:



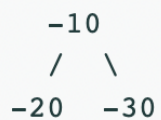
Output 1:

6

Explanation 1:

The path with maximum sum is: 2 -> 1 -> 3

Input 2:



Output 2:

-10

Explanation 2

The path with maximum sum is: -10

Optimal Strategy For A Game

You are given an array **A** of size **N**. The array contains integers and is of **even length**. The elements of the array represent **N coin** of **values V_1, V_2, \dots, V_n** . You play against an opponent in an **alternating** way.

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.

You need to determine the **maximum possible amount of money** you can win if you **go first**.

Note: Both the players are playing optimally.

Example 1:

Input:

$N = 4$

$A[] = \{5, 3, 7, 10\}$

Output: 15

Explanation: The user collects maximum value as 15(10 + 5)

Example 2:

Input:

$N = 4$

$A[] = \{8, 15, 3, 7\}$

Output: 22

Explanation: The user collects maximum value as 22(7 + 15)