

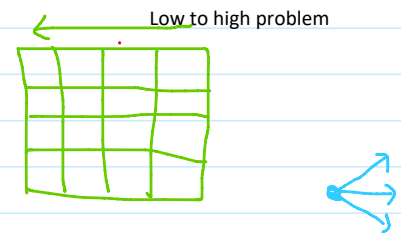
Goldmine problem

04 April 2022 20:33

<https://nados.io/question/goldmine>

```
// Goldmine problem - max. no. of golds can we mine in the path.  
//
```

```
import java.io.*;  
import java.util.*;  
  
public class Main{  
    public static void main(String[] args) throws Exception {  
        Scanner scn = new Scanner(System.in);  
        int n = scn.nextInt();  
        int m = scn.nextInt();  
  
        int[][] costs = new int[n][m];  
        for(int i=0;i<costs.length;i++)  
        {  
            for(int j=0;j<costs[0].length;j++)  
            {  
                costs[i][j] = scn.nextInt();  
            }  
        }  
  
        int[][] dp=new int[costs.length][costs[0].length];  
        for(int j=dp[0].length-1;j>=0;j--)  
        {  
            for(int i=0;i<dp.length;i++)  
            {  
                if(j==dp[0].length-1)  
                {  
                    dp[i][j]=costs[i][j];  
                }  
                else if(i==0)  
                {  
                    dp[i][j]=costs[i][j]+Math.max(dp[i][j+1],dp[i+1][j+1]);  
                }  
                else if(i==dp.length-1)  
                {  
                    dp[i][j]=costs[i][j]+Math.max(dp[i][j+1],dp[i-1][j+1]);  
                }  
                else  
                {  
                    dp[i][j] = costs[i][j]+ Math.max(dp[i-1][j+1],Math.max(dp[i][j+1],dp[i+1][j+1]));  
                }  
            }  
        }  
        int max=-1;  
  
        for(int i=0;i<dp.length;i++)  
        {  
            if(max<dp[i][0])  
            {  
                max=dp[i][0];  
            }  
        }  
        System.out.println(max);  
    }  
}
```



Problem is divided into 4 parts -

1. When column == last
2. When row == 0th
3. When row == last
4. Remaining part

Fill the 2-d dp array .

Then from the first column, find the maximum element

}
}