

# CS23331-Design and Analysis of Algorithms-2023 Batch-CSE

Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 2-DP-Playing with chessboard

## Quiz navigation



Finish review

Started on	Tuesday, 22 October 2024, 2:17 PM
State	Finished
Completed on	Tuesday, 22 October 2024, 2:46 PM
Time taken	29 mins 4 secs
Grade	10.00 out of 10.00 (100%)

### Question 1

Correct

Mark 10.00 out of 10.00

Flag question

### Playing with Chessboard:

Ram is given with an  $n \times n$  chessboard with each cell with a monetary value. Ram stands at the (0,0), that the position of the top left white rook. He is been given a task to reach the bottom right black rook position ( $n-1, n-1$ ) constrained that he needs to reach the position by traveling the maximum monetary path under the condition that he can only travel one step right or one step down the board. Help ram to achieve it by providing an efficient DP algorithm.

#### Example:

##### Input

3

1 2 4

2 3 4

8 7 1

##### Output:

19

#### Explanation:

Totally there will be 6 paths among that the optimal is  
Optimal path value:  $1+2+8+7+1=19$

#### Input Format

First Line contains the integer  $n$   
The next  $n$  lines contain the  $n \times n$  chessboard values

#### Output Format

Print Maximum monetary value of the path

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
2 int main(){
3     int n;
4     scanf("%d",&n);
5     int arr[n][n];
6     for(int i=0;i<n;i++){
7         for(int j=0;j<n;j++){
8             scanf("%d",&arr[i][j]);
9         }
10    }
11    int max(int a,int b){
12        if(a>b){
13            return a;
14        }
15        else{
16            return b;
17        }
18    }
19    int dp[n][n];
20    dp[0][0]=arr[0][0];
21    for(int j=1;j<n;j++){
22        dp[j][0]=dp[j-1][0]+arr[j][0];
23    }
24    for(int i=1;i<n;i++){
25        dp[0][i]=dp[0][i-1]+arr[0][i];
26    }
27    for(int i=1;i<n;i++){
28        for(int j=1;j<n;j++){
29            dp[i][j]=arr[i][j]+max(dp[i-1][j],dp[i][j-1]);
30        }
31    }
32    printf("%d",dp[n-1][n-1]);
33 }
34 }
```

Input Expected Got

✓	3 1 2 4 2 3 4 8 7 1	19	19	✓
✓	3 1 3 1 1 5 1 4 2 1	12	12	✓
✓	4 1 1 3 4 1 5 7 8 2 3 4 6 1 6 9 0	28	28	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 10.00/10.00.

[Finish review](#)

[← 1-DP-Playing with Numbers](#)

Jump to...

[3-DP-Longest Common Subsequence →](#)