

Problem

In this problem, we have one grid and each cell have some value and we have to start from (0,0) and reach the end but we have to minimise the sum of cell values

Approach:

→ Here we will try all the paths and then take minimum of it.
we can do it using recursion

```
int solve(int i,int j,vector<vector<int>>&arr){
    if(i == 0 && j == 0){
        return arr[i][j];
    }
    if(i < 0 || j < 0){
        return INT_MAX; // we want minimum value so that
        if it returns INT_MAX so it will be not considered
    }
    int up = arr[i][j] + solve(i-1,j,arr);
    int left = arr[i][j] + solve(i,j-1,arr);
    return min(up,left);
}
int main(){
    return solve(n-1,m-1,arr);
}
```

Memoization Code

```
int solve(int i,int
j,vector<vector<int>>&arr,vector<vector<int>>&dp){
    if(i == 0 && j == 0){
        return arr[i][j];
    }
    if(i < 0 || j < 0){
```

```

        return INT_MAX; // we want minimum value so that
if it returns INT_MAX so it will be not considered
    }
    if(dp[i][j] != -1){
        return dp[i][j];
    }
    int up = arr[i][j] + solve(i-1,j,arr);
    int left = arr[i][j] + solve(i,j-1,arr);
    return dp[i][j] = min(up,left);
}
int main(){
    vector<vector<int>>>dp(n,vector<int>(m,-1));
    return solve(n-1,m-1,arr,dp);
}

```

Tabulation Code

```

int solve(int i,int
j,vector<vector<int>>&arr,vector<vector<int>>&dp){
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(i == 0 && j == 0){
                return dp[i][j] = arr[i][j];
            }
            int up=arr[i][j],left=arr[i][j];
            if(i>0){
                up += dp[i-1][j];
            }else{
                up += INT_MAX;
            }
            if(j>0){
                down += dp[i][j-1];
            }else{
                down += INT_MAX;
            }
            dp[i][j] = min(up,down);
        }
    }
}

```

```

    }
    return dp[n-1][m-1];
}
int main(){
    vector<vector<int>>dp(n,vector<int>(m,-1));
    return solve(n-1,m-1,arr,dp);
}

```

Space optimisation

```

int solve(int i,int j,vector<vector<int>>&arr){
    vector<int>prev(m,0);
    for(int i=0;i<n;i++){
        vector<int>curr(m,0);
        for(int j=0;j<m;j++){
            if(i == 0 && j == 0){
                return curr[j] = arr[i][j];
            }
            int up=arr[i][j],left=arr[i][j];
            if(i>0){
                up += prev[j];
            }else{
                up += INT_MAX;
            }
            if(j>0){
                down += curr[j-1];
            }else{
                down += INT_MAX;
            }
            curr[j] = min(up,down);
        }
        prev = curr;
    }
    return prev[m-1];
}
int main(){

```

```
        return solve(n-1,m-1,arr,dp);  
    }
```