(1). we have the structure: if i,j is not an obstacle

Num [from $0,0 \to i,j$]

= Num [from $0,0 \to i-1, j$] + Num [from, $0,0 \to i, j-1$]

+ Num [from $0,0 \to i-1, j-1$]

if $(i-1, j)$ or $(i, j-1)$ or $(i-1, j-1)$ is the obstacle, the number is 0.

So we can compute a 2D array:

arr [i] [j] means the number of ways from $0,0 \to i,j$.

(2). pseudo-code.

```
find Num ( obstacle Arr, m, n )

    create dp [m] [n] = {0}.
    for i in 1 ~ m-1:
        if dp [i] [0] is not an obstacle:
            dp [i] [0] = dp [i-1] [0]
    for i in 1 ~ n-1:
        if dp [0] [i] is not an obstacle:
            d [0] [i] = dp [0] [i-1]
```

```
for i in 1 ~ m-1 :
    for j in 1 ~ n-1 :
        if (i,j) is an obstacle:
            dp[i, j] = 0
        else :
            dp[i, j] = dp[i-1, j] + dp[i, j-1]
                       + dp[i-1, j-1]
return dp[m-1, n-1];
```