Recursion II

What is Recursion?

The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function.

NOTE:

- Code will always go line-by-line.
- Each function variables are stored or preserved for each call.
- In recursion, ‘function calls’ are stored in a type of data structure called stack separately.

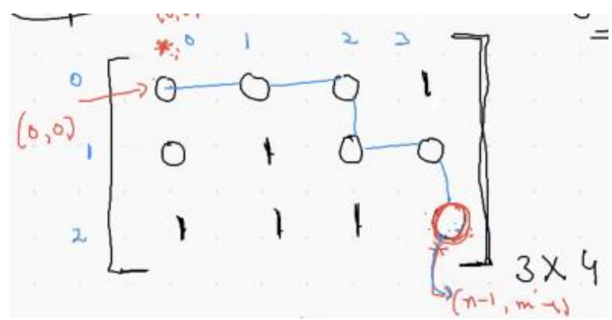
What is base condition in recursion?

In the recursive program, the solution to the base case is provided and the solution of the bigger problem is expressed in terms of smaller problems.

How to think recursively?

A matrix, m X n is of 0’s and 1’s. there will be either 0 or 1 in the matrix.

$$
\begin{matrix}
0_{(0,0)} & 0_{(0,1)} & 0_{(0,2)} & 1_{(0,3)} \\
0_{(1,0)} & 1_{(1,1)} & 0_{(1,2)} & 0_{(1,3)} \\
1_{(2,0)} & 1_{(2,1)} & 1_{(2,2)} & 0_{(2,3)}
\end{matrix}
$$



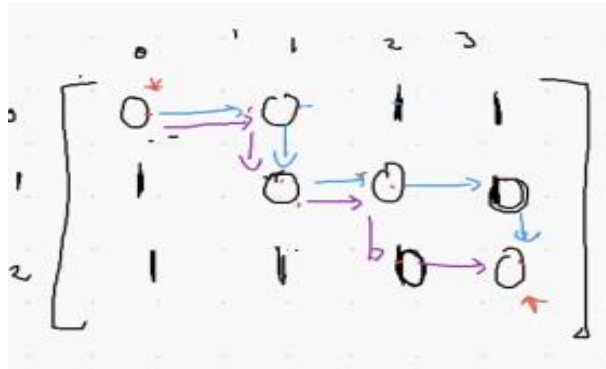
This matrix has 3 rows and 4 columns. In this matrix, 1 denotes a wall and 0 denotes empty space. Let’s say a rat at (0,0) wants to reach (n-1, m-1).

Find number of ways to find number of ways in which rat can reach (n-1, m-1) cell from (0,0) cell if rat can move in **2** directions (right & down).

There is only one way through which the rat can reach its destination.

IN the below example, rat has two ways to reach its destination. Loops cannot be used to solve such situation. If you are not able to think about solutions easily, then recursion is a good way.

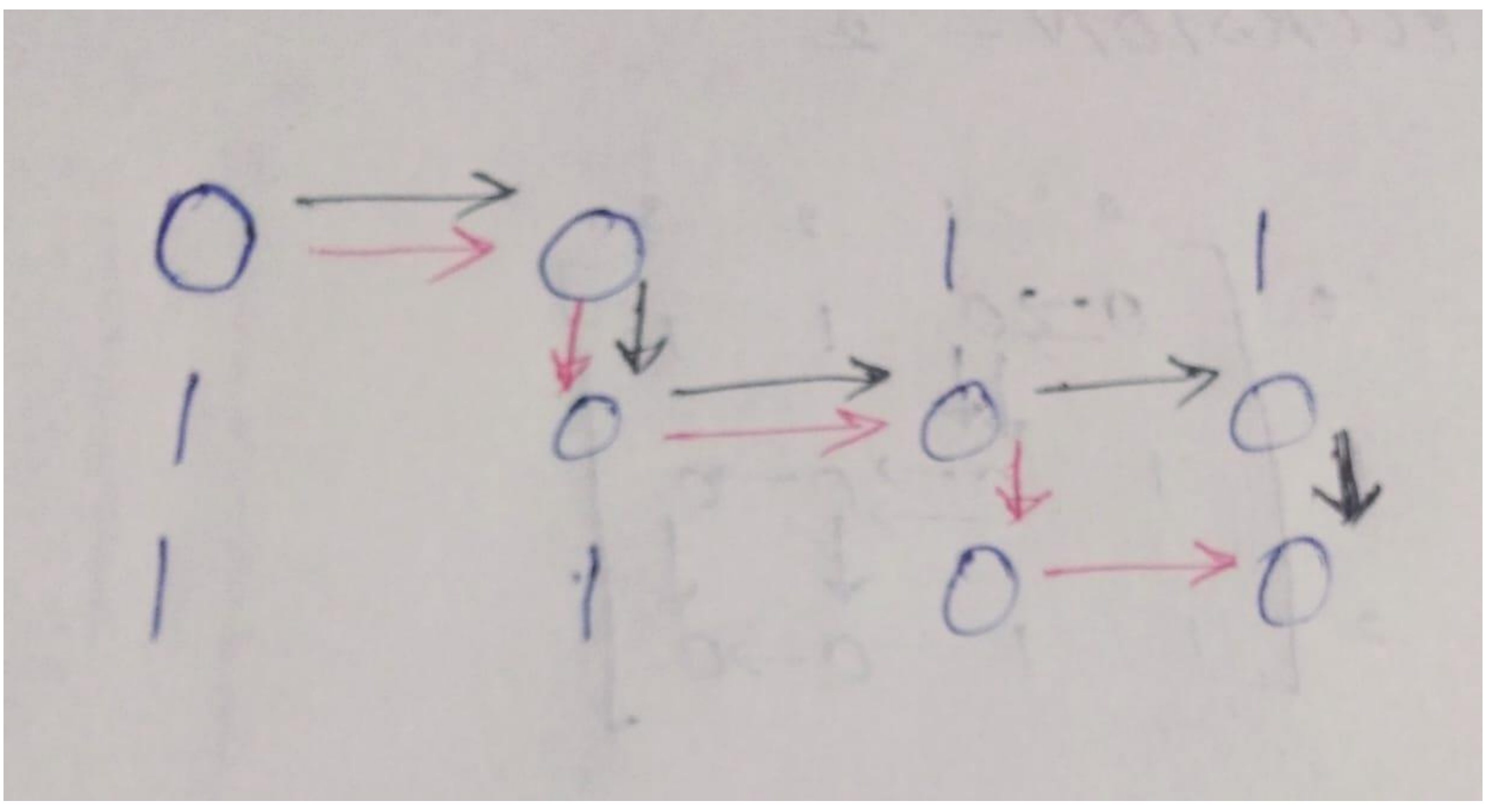
$$
\begin{array}{cccc}
0_{(0,0)} & 0_{(0,1)} & 1_{(0,2)} & 1_{(0,3)} \\
1_{(1,0)} & 0_{(1,1)} & 0_{(1,2)} & 0_{(1,3)} \\
1_{(2,0)} & 1_{(2,1)} & 0_{(2,2)} & 0_{(2,3)}
\end{array}
$$



*In **any problem**, think about **choices**.*

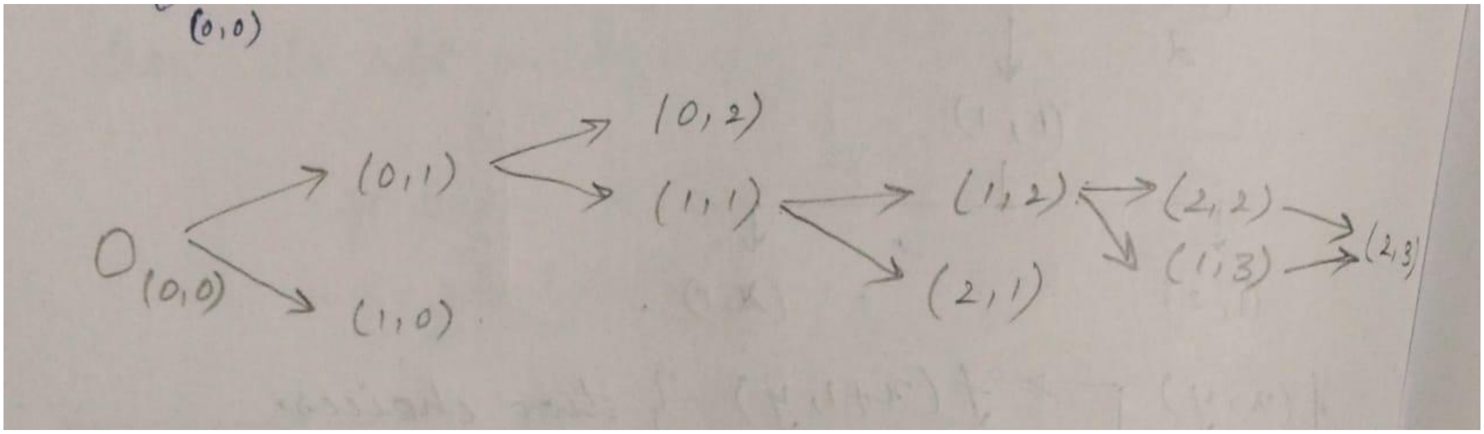
NOTE: whenever you encounter a recursion problem, think what choices you have.

Rat is at (0,0) and the rat can move in right and downward direction. If the rat moves in the right direction, then it’s position would be (0,1). It can move downwards from (0,1)



From (0,1) the rat has two choices, it moves towards (0,2) or (1,1). Since (0,2) is a wall, it will move towards (1,1). From this point, it can move to (1,2) or (2,1). As (2,1) is wall, it will go to (1,2).

At (1,2), there is no wall and hence it can move in any of the two directions. So, the rat now goes in down to (2,2) or to right (1,3). IN any case, it would one step away from its destination (2,3)



If this returns to the parent 1, we can sum them up. So, we are returning 1 for each path that the rat used to reach its destination.

If there are x ways to do something and y ways to do something, then total no. of ways to do something is x + y.

In this question, I have one choice towards right and one towards left. So, my total number of solutions that I have is right + left. With this in mind,

F (x, y) is a function which tells number of ways to reach (n-1, m-1).

$F(x +, y)$ OR $F(x, y + 1)$ are two ways to reach the destination.

OR means adding
AND means multiplication.

$$\text{So, } F(x, y) = F(x+1, y) + F(x, y+1)$$

If a person has 2 ties and 3 pairs of socks, then he can wear them in 6 ways.

If a person has 2 red and 3 blue shoes, he can wear them in 5 ways.

CODE:

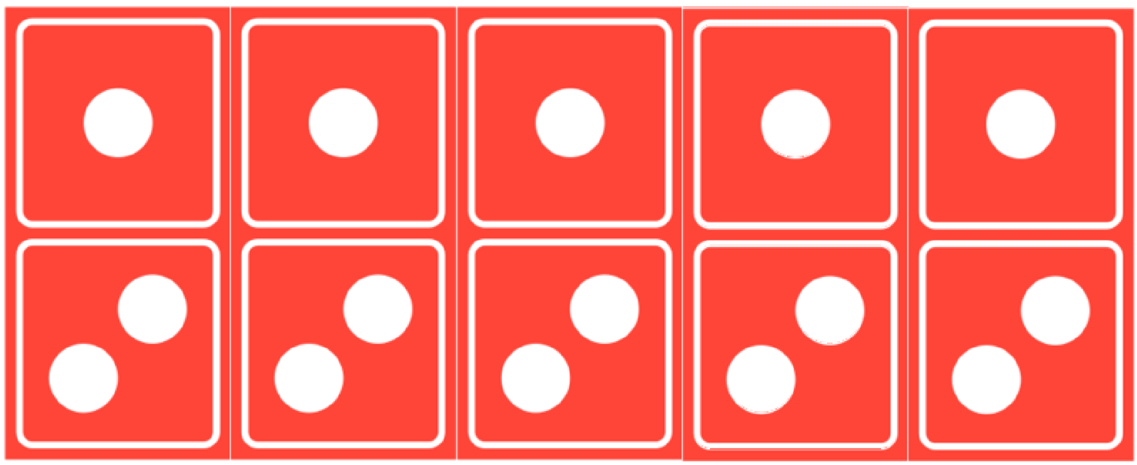
```python
"""
Given a maze, a rat is at starting position (0, 0) we have to find the
no. of ways in which it can reach destination
"""

def NoOfWays(x, y, n, m, grid):

    if x >= n:
        return 0

    if y >= m:
        return 0

    if grid[x][y] == 1: # if there is a wall
        return 0

    if x == n - 1 and y == m - 1:
        return 1

    ans = NoOfWays(x, y + 1, n, m, grid) + NoOfWays(x + 1, y, n, m, grid)
    return ans

if __name__ == "__main__":

    grid = [[0, 0, 1, 1], [1, 0, 0, 0], [1, 1, 0, 0]]
    n = len(grid)
    m = len(grid[0])

    print(NoOfWays(0, 0, n, m, grid))
```

Q) Dominos, given a rectangle of dimension N X 2 and you have a domino of dimension 2 X 1. IN how many ways can you tile the rectangle.

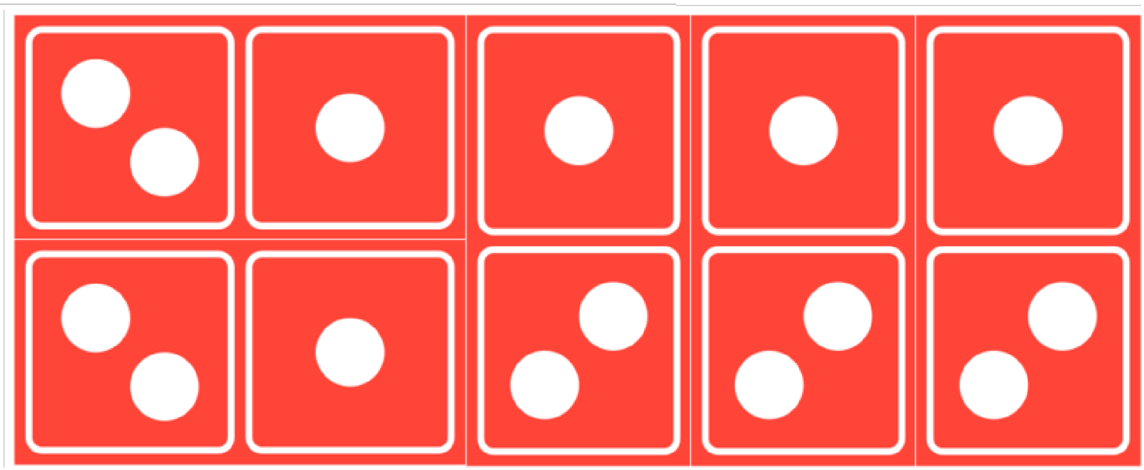
We have a 2 X 1 dominos. The height of the dominos is 2 and width is 1.

Now, in rectangle, of 2 X n dimension, in how many ways can you tile to form a rectangle.

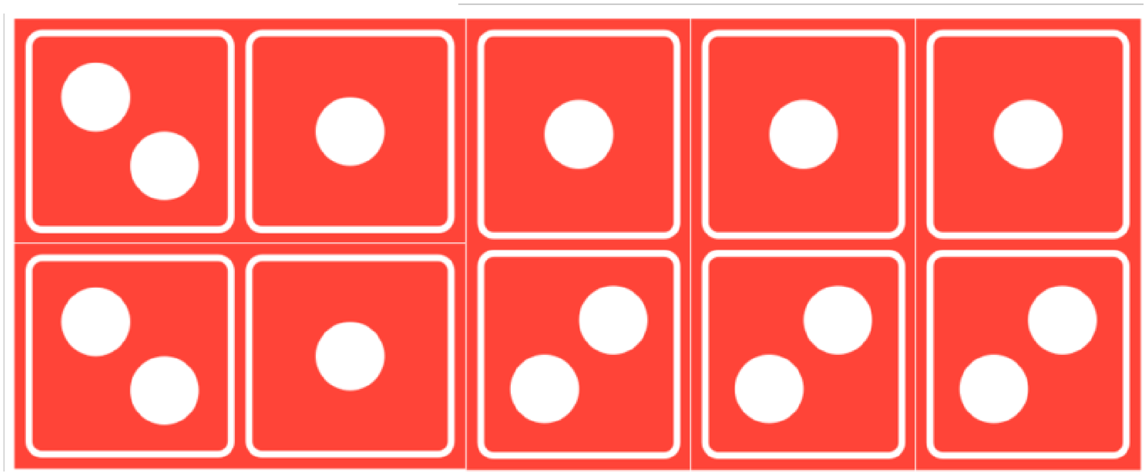
1



2



3

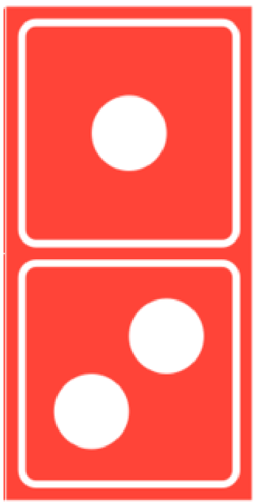
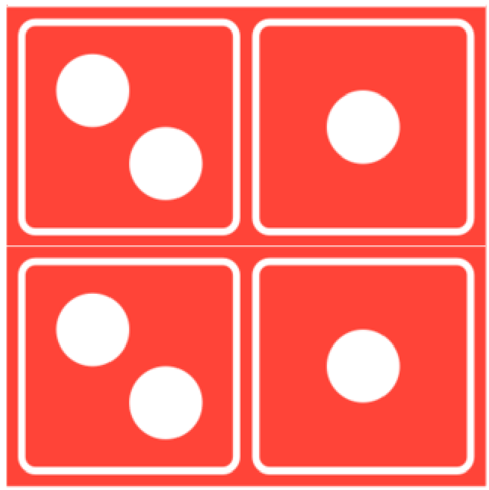


So, there can be many ways in which we can tile this rectangle with a domino.

Given a domino of dimension n X 2 and you have a domino of dimension 2 X 1. In how many ways can you tile the rectangle.?

Let’s suppose a rectangle of 2 X 3 using dominos 2 X 1. The choice with this dominos would be,

I can lay the domino vertically or horizontally. It we a lay it vertically, we can place one domino, but if we a placing it horizontally, then we have to have place 2 dominos so that the length is equal to 2.

 vertical case,  horizontal case

If the length was N and it, we place to dominos horizontally, then will reduce to N – 2. It we lay domino vertically then N will be reduced to N – 1 and when N <= 0, then we have our solution. We can derive:

$$F(N) = F(N-1) + F(N-2)$$

CODE:

```
def Tiling(N):
    if N < 0:
        return 0

    if N == 0:
        return 1

    ans = Tiling(N - 1) + Tiling(N - 2)
    return ans

print(Tiling(5))
```