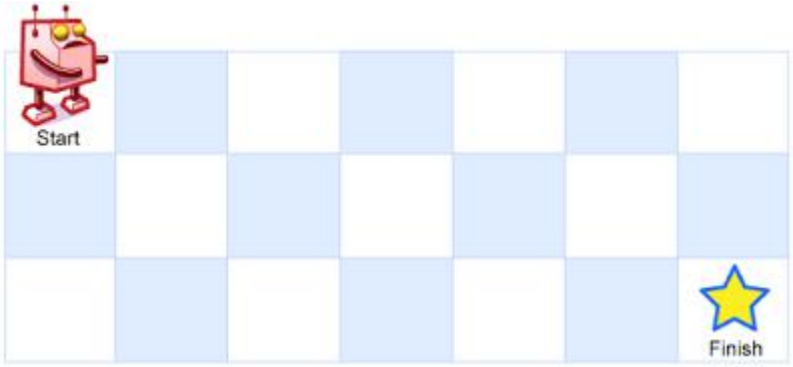
DAY 43

INTERVIEW BIT PROBLEMS :

1. Grid Unique Paths

A robot is located at the top-left corner of an A x B grid (marked 'Start' in the diagram below).



The robot **can only move either down or right at any point in time**. The robot is trying to reach the bottom-right corner of the grid (marked 'Finish' in the diagram below).
How many possible unique paths are there?
Note: A and B will be such that the resulting answer fits in a 32 bit signed integer.
Example :
Input : A = 2, B = 2
Output : 2

2 possible routes : (0, 0) -> (0, 1) -> (1, 1)
OR : (0, 0) -> (1, 0) -> (1, 1)

CODE :

PYTHON

```
class Solution:
    # @param A : integer
    # @param B : integer
    # @return an integer
    def uniquePaths(self, A, B):
        if A==1 or B==1:
            return 1
        u_paths=[[0 for i in range(B)]for j in range(A)]
        for i in range(1,A):
            u_paths[i][0]=1
        for j in range(1,B):
            u_paths[0][j]=1
        for i in range(1,A):
            for j in range(1,B):
                u_paths[i][j]=u_paths[i][j-1]+u_paths[i-1][j]
        return u_paths[A-1][B-1]
```

C++

```
int Solution::uniquePaths(int A, int B) {
    vector<vector<int>> grid(A,vector<int>(B,0));
    if(A==0 && B==0)
        return 1;
    grid[0][0]=1;
    for(int i=0;i<A;i++){
        for(int j=0;j<B;j++){
            if(i-1>=0)
                grid[i][j] += grid[i-1][j];
            if(j-1>=0)
                grid[i][j] += grid[i][j-1];
        }
    }
    return grid[A-1][B-1];
}
```