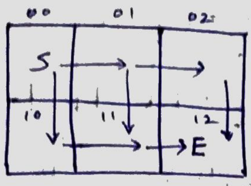
Grid Unique Path

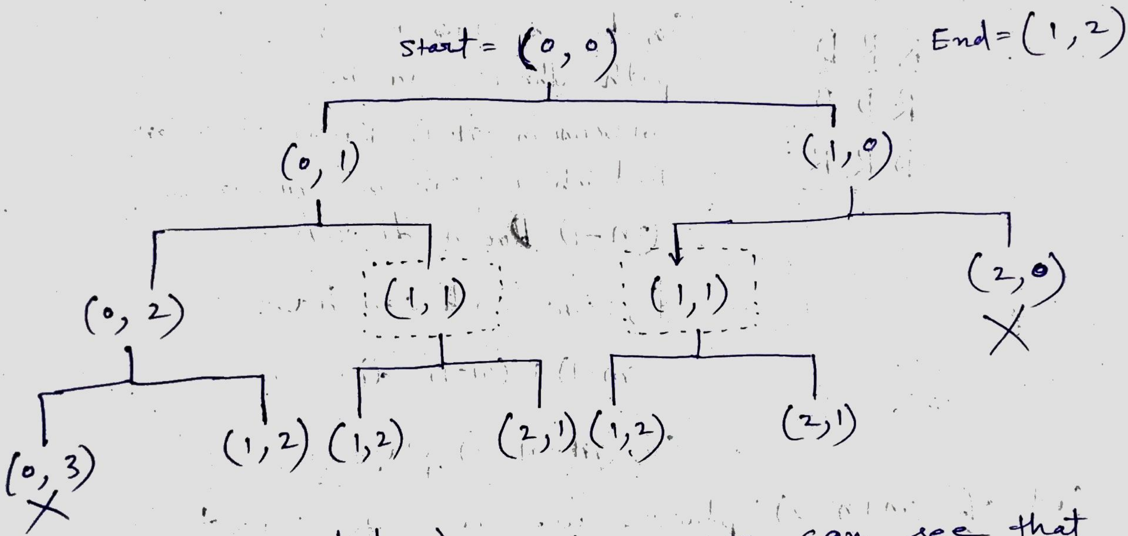


R R D
R D R
D R R
} 3 ways

Using recursion we can find different all possible ways.

Recursive Equation	Start = (i, j) End = (p, q)

$$\text{UniquePaths}(i,j) = \begin{cases} \text{UniquePaths}(i, j+1) + \text{UniquePaths}(i+1, j) & ; i <= p \,\&\&\, j <= q \\ 1 & ; i == p \,\&\&\, j == q \\ 0 & ; i > p \,\|\, j > q \end{cases}$$



T = Exponential
S = Exponential

But, here, we can see that some of sub problems are overlapping

So, we can use Dynamic Programming

We can use a matrix to store the value of ~~already~~ calculate sub-problems.
And use the stored result whenever we encounter the same sub-problem in future.

00 S	01	02
10	11	12 E

dp

00 -1	01 -1	02 -1
10 -1	11 -1	12 -1

Initialize the dp table with -1.

Using DP :-

$T = O(mn)$

$S = O(mn)$

But, we have even better solution than DP sol.
Using the concept of $\boxed{\text{Combinations}}$ we can solve this question.

R R D
R D R
D R R

So, here ~~one~~ in one path there can be maximum $(n-1)$ Right directions. And there can be maximum $(m-1)$ Down directions.

So, one path can have $(m-1) + (n-1)$ directions.

So, $(m+n-2)$ places.

Out of $(m+n-2)$ places we have to select $(m-1)$ down direction or $(n-1)$ right directions.

So, No. of possible paths $= {}^{m+n-2}C_{m-1}$ or ${}^{m+n-2}C_{n-1}$

Now, we already know that ${}^{N}C_{R}$ can be found in $T = O(R)$

So, Our $T = O(\min(m, n))$, $S = O(1)$

Now, this concept can even be extended if there is a matrix of $m \times n$ size and starting position is (i, j) and ending position is (p, q) where $\underline{i <= m-1}$ && $\underline{j <= n-1}$.

i.e., (i, j) and (p, q) are random positions in a matrix of size $\underline{m \times n}$.

Time Complexity in case of DP $= O(p-i * q-j)$
Space complexity in case of DP $= O(mn)$

Time Complexity in case of combination $= O(\min(p-i, q-j))$

$^{(p+q)-(i+j)}C_{(p-i)}$ or $^{(p+q)-(i+j)}C_{(q-j)}$

Space complexity in case of combination $= O(1)$