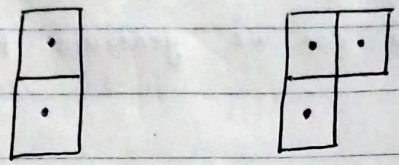
790: Domino & Tromino Tiling

Input: $2 \times n$ (n is input)

↳ matrix

We have ~~two~~ dominos and tromino



(can be rotated)

Q:- Find the number of ways by which $(2 \times n)$ matrix can be filled with dominos and trominos.

I tried to solve this question using tree based structure. but I was not able to create a proper tree to solve it.

I thought that the question can be translated to something like this.

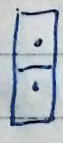
$2 \times n$ boxes, and we have 2, 3. Find the number of ways 2 and 3 can be summed to reach the target of $2 \times n$. But this doesn't take rotation into account.

Another approach was

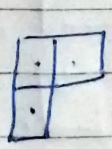
def func(row1, row2)

Here we start with row1 = n and row2 = n, and subtract from row1 and row2 based on the domino and tromino orientation.

Like.

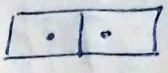


⇒ row1 - 1
row2 - 1



row1 - 2
row2 - 1

Do the same for the other orientation as well

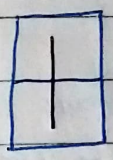
But this fails as well because  this orientation of domino can be placed either in row 1 or row 2. (Try it, this logic doesn't work)

<u>Lesson learnt</u>:- When you are not able to create a tree to solve the dp problem. Fallback to tabulation approach directly. See how can we generate F(n) based on previous values.

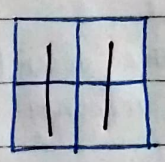
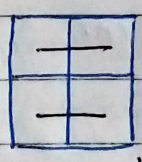
<u>This can also be done in two ways</u>

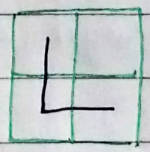
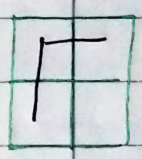
① Assume you have values till i
$F(i)$ and how $F(i+1)$ can be calculated.

② Here try to play around and calculate
$F(1)$, $F(2)$, $F(3)$. Assumption $F(i)$, doesn't come straightforward here. (Used Here)

$f(1)$ = 

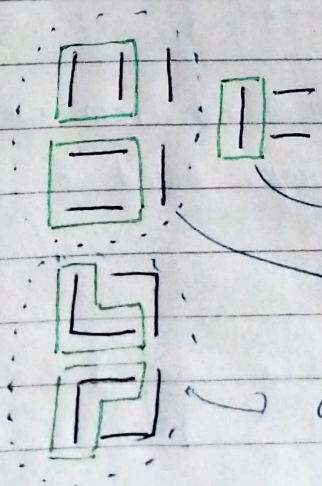
~~$f(2)$~~ =

$f(2)$ =  

$p(2)$  

fully covered (fc)

partially covered (pc)
last column not completely filled
(symmetric)
∴ we will count this as 1

$f(3)$ = 

created from $f(1)$

created from $f(2)$

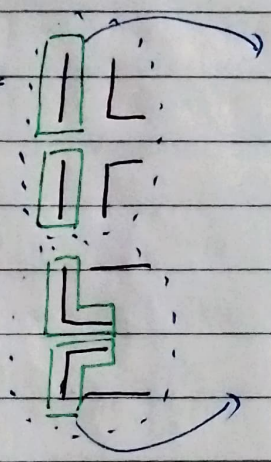
created from $p(2)$

We see that we created $f(3)$ from $f(2)$, $f(1)$ and $p(2)$

$$\therefore \boxed{f(n) = f(n-1) + f(n-2) + 2 * p(n-1)}$$

↳ multiplying with two because of symmetry

We have a recursive function for the $f(n)$. But since it depends on function 'p'.
We also need a recursive function for $p(n)$.

$p(3) =$ 

| comes from $f(1)$

This comes from $p(2)$

$\therefore$ Partially covered (pc) comes from adding tromino to $f(n-2)$ and adding horizontal domino to $p(n-1)$

$$\therefore \boxed{p(n) = f(n-2) + p(n-1)}$$

Now we have recursive functions. we can used this to solve in $O(n)$ time (In one for loop).

Also in $O(1)$ space complexity, since values of $f(n)$ and $p(n)$ depends only on last two values.