(*) Catalan Number

Recursive def $\rightarrow \begin{cases} C_n = \sum_{i=0}^{n-1} C_i C_{n-i-1} \\ C_0 = 1 \end{cases}$

Ex $\rightarrow$ $C_1 = C_0 C_0$

$C_2 = \text{l.o.l} \sum_{i=0}^{1} C_i C_{2-i-1} = C_0 C_1 + C_1 C_0$

$C_3 = C_0 C_2 + C_1 C_1 + C_2 C_0$

$C_4 = C_0 C_3 + C_1 C_2 + C_2 C_1 + C_3 C_0$

- - - - - - -

- - - - - - - - -

Formula based on binomial coefficient $\rightarrow$

$$C_n = {}^{2n}C_n - {}^{2n}C_{n+1}$$

$$= \binom{2n}{n} - \binom{2n}{n+1}$$

$$= \frac{2n!}{n!\,n!} - \frac{2n!}{(n+1)!\,(2n-n-1)!}$$

$$= \frac{2n!}{n!\,n!} - \frac{2n!}{(n+1)!\,(n-1)!}$$

$$= \frac{2n!}{n!\,n!} - \frac{2n! \times n}{(n+1)\cdot n! \times n!}$$

$$= \frac{2n!}{n!\,n!}\left(1 - \frac{n}{n+1}\right)$$

$$= \frac{2n!}{n!\,n!\,(n+1)} \Rightarrow \frac{2n!}{n!\,(n+1)!}$$

Formula for
Catalan Number

$$\boxed{C_n = {}^{2n}C_n - {}^{2n}C_{n+1} = \frac{2n!}{(n+1)!\cdot n!}}$$

First few Catalan Number.

$n =$	0	1	2	3	4	5	6	7 ... are
	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
	1	1	2	5	14	42	132	429

Catalan (n) $\longrightarrow$ from (n-1) th Catalan

$$C(n) = C_{n-1} \times \left(\frac{4n-2}{n+1}\right)$$

$$C(n) = \frac{2n!}{(n+1)!\cdot n!}$$

$$\boxed{\left(\frac{4n-2}{n+1}\right)}$$

$$C(n-1) = \frac{2(n-1)!}{(n-1+1)\,(n-1)!}$$

$$\frac{C_n}{C_{(n-1)}} = \frac{\frac{2n!}{(n+1)!\,\cancel{n!}}}{\frac{\cancel{2n}!\,2(n-1)!}{(\cancel{n})!\,(n-1)!}} = \frac{2n!\times(n-1)!}{(n+1)!\,(2n-2)!} = \frac{2\cancel{n}\times 2n-1}{(n+1)\,(\cancel{n})}$$

Implementation (Catalan Number)

I/P: $n=3 \rightarrow$ O/P $\rightarrow 5$.

1. Using formula $C_n = \frac{2n!}{(n+1)!\,n!}$

```
int fact (int n)
{ int fact = 1;
  for (int i= n; i>0; i--){
        fact = fact * i;
  }
  return fact;
}
int main ()
{
  int n;
   cin>>n;
   cout<<"nth cat no." << (fact (2*n))/(fact(n+1) *
                                   fact(n));
   return 0;
}
```

SC = O(1)
TC = O(n). or O(3n)

② Using Recc:

```
int catNo (int N){
 if (n==0) return 1;
   int cat=0;
 for (int i=0; i<=n-1 ; i++)
      cat += CatNo(i) * Cat (N-i-1);
 return cat;
}
```

TC → exponential

Overlapping sub problem → Dp memo
→ Dp tabu

memo

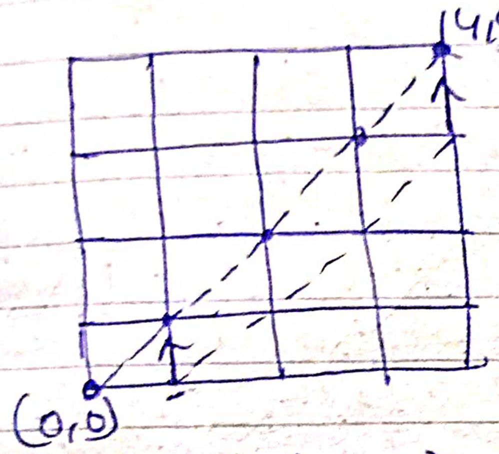
dp(n+1).

```
int CatNo(int N, vector <int>& dp)
{
              dp[0]= return dp[0];
  if(n==0) ~~return 1;~~ ~~return 1;~~
  if( dp [n] !=-1) return dp[n];

  for (int i=0; i<=n-1 ; i++)
    ~~cat~~ dp [n] = CatNo (i, dp) * Cat (N-i-1, dp)

 return dp[n].
 }.
```

formula → $\boxed{Catalan(n) = \frac{1}{n+1}\ {}^{2n}C_n}$

Cpp → ncr(2n, n)/n+1

(*) Count ways to Reach a Grid Top.



n=4

(0,0) → (4,4)

↑ → ✓

dont cross diag

(1,1) → (1,2) →

(0,0) → (0,1) → (0,2)

C#A.

① if ~~(i==j)~~ s(i, j+1) ~~...~~
if (i<j & i=j) s(i, j+1) + s(i+1, j+1).

((0,0) to (i,j)) * ((i,j) to des(4,4)).

$C(4) = C(0)*C(3) + C(1)*C(2) + C(2)\,C(1) + C(3)*C(0)$

$$C(n) = \sum_{i=0}^{n-1} C(i) * C(n-i-1)$$

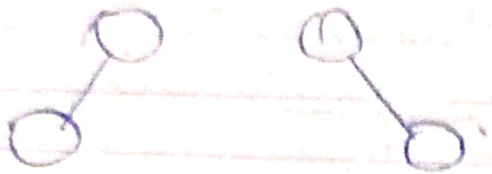
$$= \frac{2n!}{(n+1)!\,(n)!}$$

$\boxed{C(0)=1}$

No of BT

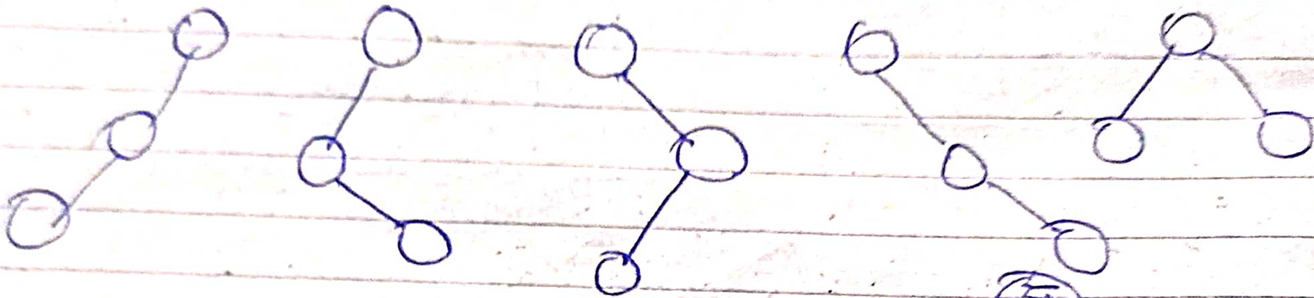
↳ Catalan Number

n=2 #BT unlabelled



2

n=3 #BST



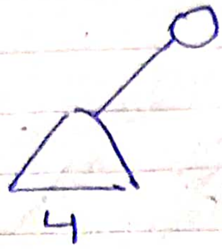
5

n=1 ↳ O → 1

n=0 ↳ $\emptyset$ — 1.

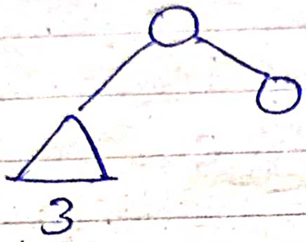
for n=5

Recur



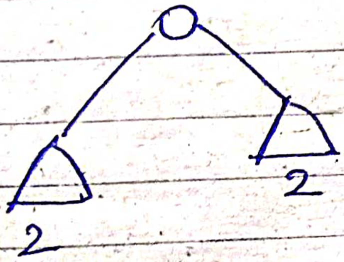
$C(5) = C(4) * C(0)$

$+$



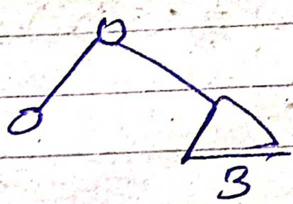
$C(3) * C(1)$

$+$



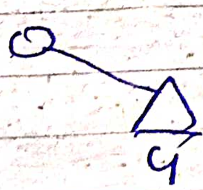
$C(2) * C(2)$

$+$



$C(1) * C(3)$

$+$



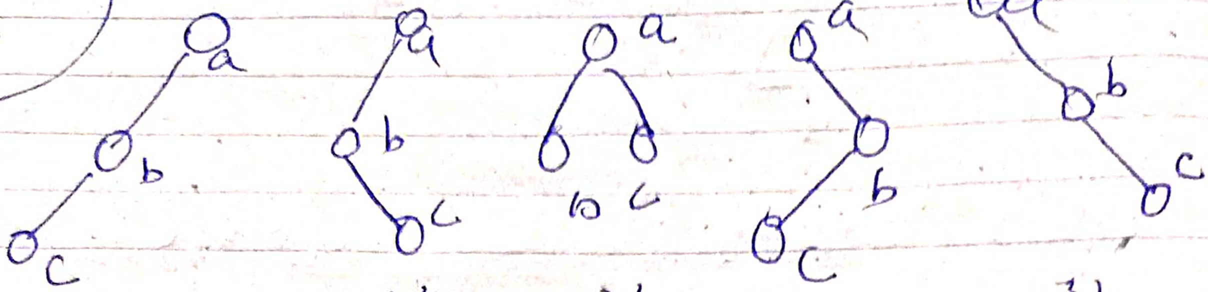
$C(0) * C(4)$

This is for unlabeled

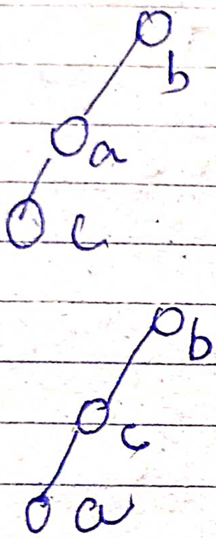
for labeled $n!$ times unlabeled.

labeled BT

n=3 !



3!   3!   3!   3!   3!



$$\boxed{n! \times C(n)}$$

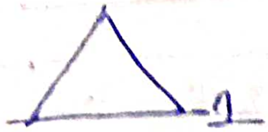
RR → $n! \times C(n)$ $\begin{cases} \sum_{i=0}^{n-1} C_i C_{n-i-1} \\ C_0 = 1 \end{cases}$

→ $\cancel{n!} \times \frac{2n!}{(n+1)!\,\cancel{n!}} = \boxed{\frac{2n!}{(n+1)!}}$

labeled BT

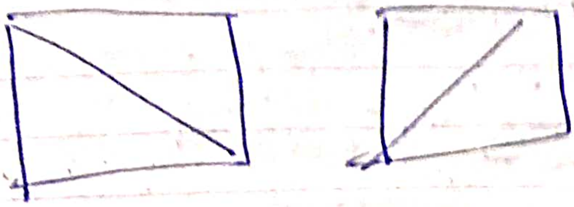
Counting Polygon Triangulations

n = 3 :



1

n = 4 :

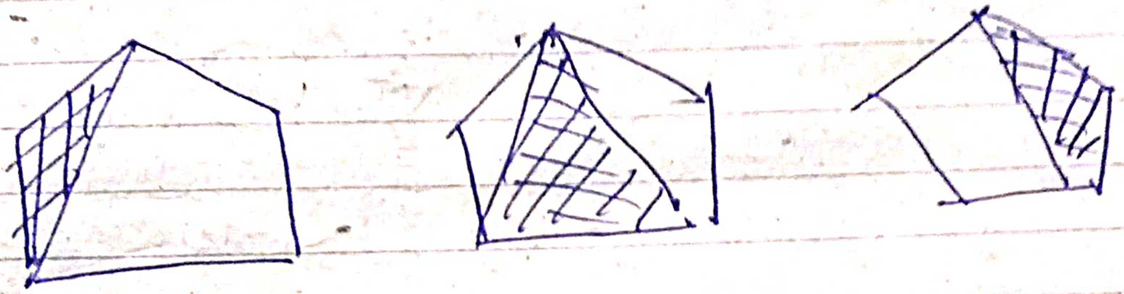


$C_2 = 2$.

$C_2 = 2$

Ans → $\boxed{C(n-2)}$.

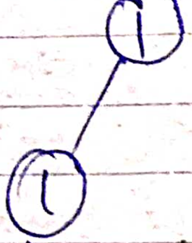
n=5

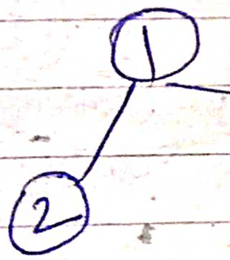


n=5

$C_3 = C_0 C_2 + C_1 C_1 + C_2 C_0$.

X





→ $\underline{2! \times C(2)}$

labeled

$2! \times C(n)$.

perm

$\frac{2! \times C(2)}{2!}$

$\frac{P\#}{9/10}$