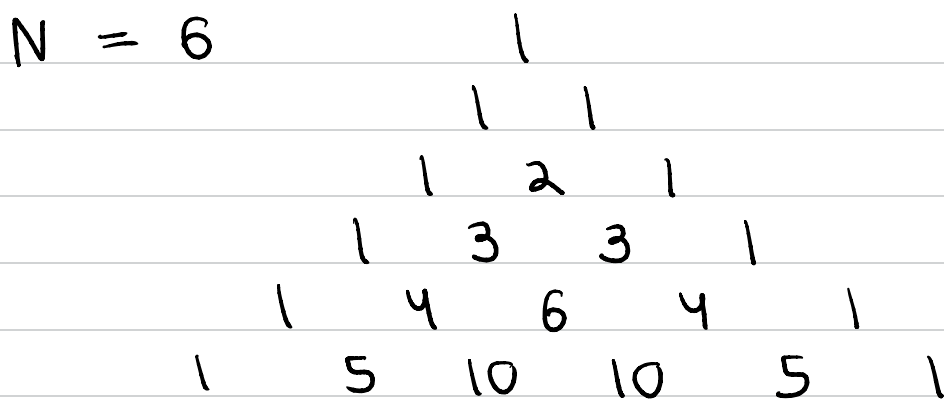


PASCAL'S TRIANGLE

★ In this problem we are supposed to compute nC_r in shortest possible time. or just print a specific row/element of the triangle. A Pascal Triangle looks something like :



There can be 3 question variants.
1) Print a specific element

For row R and column C element is given by ${}^{R-1}C_{C-1} = \frac{(R-1)!}{(C-1)!(R-C-2)!}$

Pseudocode :

$nCr(n, r)$ {

res = 1

for (i = 0 → r) {

res = res × (n - i)

res = res / (i + 1)

}

return res

}

2) Print N^{th} row

Brute force solution here is to find all elements of this N^{th} row using the above formula

Optimal solution is to carry on the multiplication of previous element and to only multiply the extra factor

Eg :

Column :	0	1	2	3	4	5
Row 5 :	1	5	10	10	5	1

Diagram illustrating the calculation of the 5th row elements using the optimal solution (multiplying the previous element by the extra factor):

- Column 0: 1
- Column 1: 5 (calculated as 1×5)
- Column 2: 10 (calculated as $5 \times \frac{4}{2}$)
- Column 3: 10 (calculated as $10 \times \frac{3}{3}$)
- Column 4: 5 (calculated as $10 \times \frac{2}{4}$)
- Column 5: 1 (calculated as $5 \times \frac{1}{5}$)

OR

$$\text{ans} = \text{ans} \times \frac{(\text{row} - \text{col})}{\text{col}}$$

Pseudocode

pascals Triangle Row(N) {

ans = 1

print(ans)

for (i = 1 → N) {

ans = ans × (N - i)

ans = ans / i

print(ans)

}

}

3) Print Pascals Triangle

Optimal solution is to just loop from 1 to N and use the print row function each time