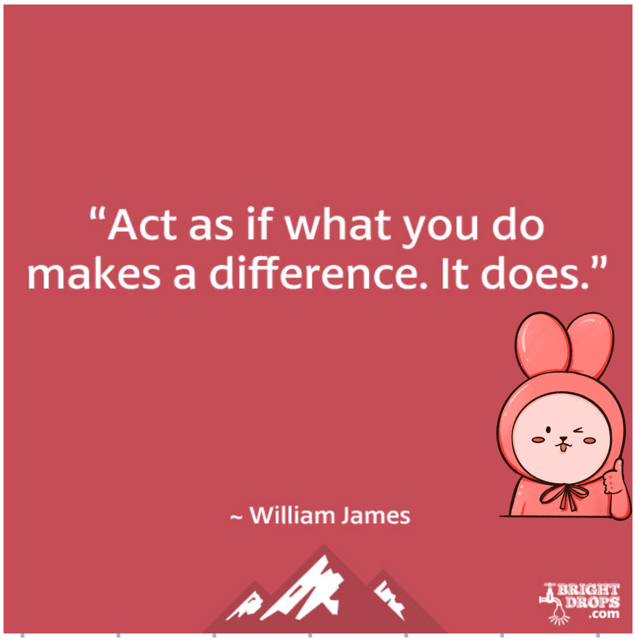
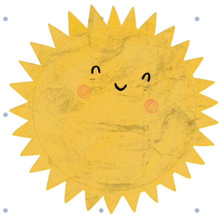
NOTES: COMBINATORICS





Good
Evening

Todays content

→ Addition & Multiplication rule

→ Permutation basics

→ Combination basics

→ Pascal Triangle

→ Find n^{th} coloumn Title

Addition & Multiplication

01. Given 3 True/False questions, every question has to be answered. In how many ways can you answer all the questions.

$$\underline{2} * \underline{2} * \underline{2} = 2^3 = 8 \text{ ways}$$

T F T F T F

F	F	F
F	F	T
F	T	F
T	F	F
T	T	F
T	F	T
T	T	T
F	T	T

8 ways

Q2 Given 10 girls & 7 boys. How many different pairs can we form?

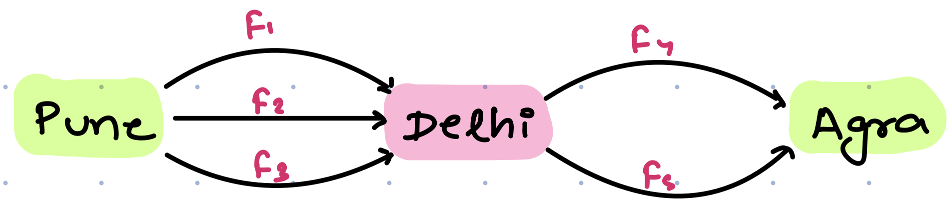
Note :- pair → 1 Girl + 1 Boy

Boys	Girls
B_1	G_1
B_2	G_2
B_3	G_3
B_4	G_4
B_5	.
B_6	.
B_7	.
	G_{10}

1 Boy AND 1 Girl

7 * 10 = 70 ways

Eg 3



No. of ways to reach Agra from Pune via Delhi

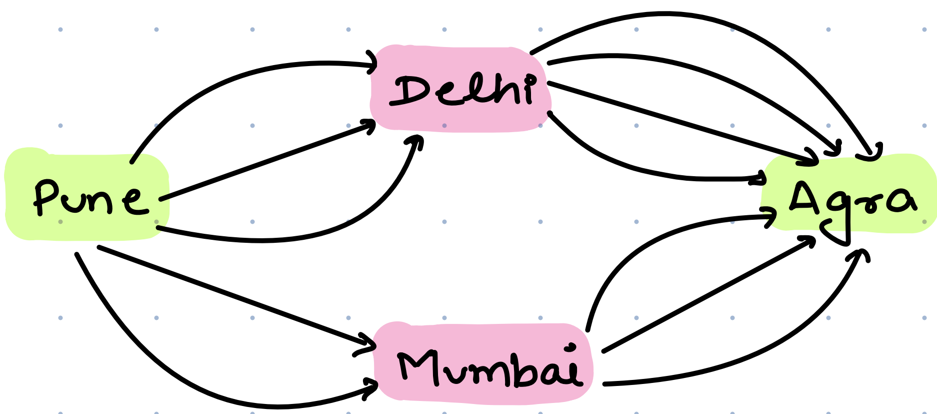
Pune → Delhi **AND** Delhi → Agra

$\underbrace{3}$ $*$ 2 = **6 ways**

F_1 F_4 F_2 F_4 F_3 F_4

F_1 F_5 F_2 F_5 F_3 F_5

Eg 4



No. of ways to reach Agra from pune ?

via Delhi OR **via Mumbai**

Pune → Delhi **AND** Delhi → Agra

$\underbrace{3 * 4}$

Pune → Mumbai **AND** M → Agra

$\underbrace{2 * 3}$

12 + 6

= **18 ways**

AND → * ⇒ used to count possibbilities that occur together in sequence

OR → + ⇒ used to count possibilites that occur in seperate ways

Scenerio

Zomato, features an exciting option for its users - meal combos. Each combo includes one main course, one *dessert*, and one *beverage*, offering a complete dining experience from various restaurants. Zomato believes that a greater variety of combos can significantly enhance customer satisfaction.

Problem

You're tasked with helping **Zomato** identify which restaurant offers the most variety in its meal combos. You're provided with a list, shaped like a grid or a 2D matrix **A**, where each row corresponds to a different restaurant's offerings.

Each row is divided into three parts:

1. A[i][0] tells you the number of main courses,
2. A[i][1] the number of desserts, and
3. A[i][2] the number of beverages a restaurant offers.

Your challenge is to analyze this data and pinpoint which restaurant gives its customers the most options to mix and match their meal combo.

```
A = [
    [3, 2, 2],    # Restaurant 1
    [4, 3, 3],    # Restaurant 2
    [1, 1, 1]     # Restaurant 3
]
```

} 3 * 2 * 2 ⇒ 12

} 4 * 3 * 3 = 36

} 1 * 1 * 1 = 1

Ans = 2nd restaurant will provide maximum no. of combos.

* Permutation

→ Arrangement of objects in $\underbrace{\text{specific order}}_{(i,j) \,!= (j,i)}$

* Given 3 distinct characters, In how many ways we can go & arrange them.

str = "a b c"

$$\underline{3} * \underline{2} * \underline{1} = 3! = 6 \text{ ways}$$

a → b → c
a → c → b
b → a → c
b → c → a
c → a → b
c → b → a

* In how many ways N distinct characters can be arranged

$$N * (N-1) * (N-2) * \ldots * 1 = N!$$

* Given 5 distinct characters, in how many ways we can arrange 2 characters

str = " a b c d e "

$\underline{5} * \underline{4} \Rightarrow 20$ ways

* Given N distinct characters, in how many ways we can arrange 3 characters?

$\underline{N} * \underline{N-1} * \underline{N-2} = N * (N-1) * (N-2)$ ways

* Given N distinct characters, in how many ways we can arrange r characters?

$\underline{N} * \underline{(N-1)} * \underline{(N-2)} * \cdots * \underline{N-(r-1)}$

$$\frac{N * (N-1) * (N-2) * \cdots * (N-r+1) * (N-r) * (N-r-1) * \cdots 1}{(N-r) * (N-r-1) * \cdots 1}$$

$= \boxed{\dfrac{N!}{(N-r)!} = {}^{N}P_{r}}$ $\Rightarrow$ No. of ways to arrange r places using N distinct characters

* **Combination** → No. of ways to select something

→ Order of selection doesn't matter $(i, j) = (j, i)$

Q Given 4 players, count the no. of ways to select 3 players

$\{P_1 \quad P_2 \quad P_3 \quad P_4\}$

$$\left.\begin{matrix} P_1 & P_2 & P_3 \\ P_1 & P_3 & P_4 \\ P_2 & P_3 & P_4 \\ P_1 & P_2 & P_4 \end{matrix}\right\} \text{4 ways}$$

Arrangements

P_1	P_2	P_3	P_1	P_3	P_4	P_2	P_3	P_4	P_1	P_2	P_4
P_1	P_3	P_2	P_1	P_4	P_3	P_2	P_4	P_3	P_1	P_4	P_2
P_2	P_1	P_3	P_3	P_1	P_4	P_3	P_2	P_4	P_2	P_1	P_4
P_2	P_3	P_1	P_3	P_4	P_1	P_3	P_4	P_2	P_2	P_4	P_1
P_3	P_1	P_2	P_4	P_1	P_3	P_4	P_2	P_3	P_4	P_1	P_2
P_3	P_2	P_1	P_4	P_3	P_1	P_4	P_3	P_2	P_4	P_2	P_1

$P_1 \; P_2 \; P_3$	$P_1 \; P_3 \; P_4$	$P_2 \; P_3 \; P_4$	$P_1 \; P_2 \; P_4$

Selections

For 1 selection = 6 arrangements

No. of selection * No. of arrangements for every selection = Total no. of arrangements

$x * 6 = 24$

$x = \frac{24}{6} = 4$

* Given n distinct elements, in how many ways we can go & select r elements

// Given n distinct ele, arrange r elements $\Rightarrow {}^nP_r = \frac{n!}{(n-r)!}$

Arrange r elements $\Rightarrow r!$

No. of selections for r distinct elements out of r elements $\Rightarrow \underline{1}$

Arrangements		selection
$r!$	$\Rightarrow$	$\underline{1}$
$\frac{n!}{(n-r)!}$	$\Rightarrow$	x

$$x * r! = \frac{n!}{(n-r)!}$$

$$x = \frac{n!}{r!\,(n-r)!}$$

Given n distinct elements, select r elements,

$$\boxed{{}^{N}C_{r} = \frac{N!}{r!\,(N-r)!}}$$

$r!$ arrangements $=$ 1 selection

1 arrangement $= \frac{1}{r!}$ selections

${}^{n}P_{r}$ arrangements $= \frac{{}^{n}P_{r}}{r!}$ selections

$= \frac{n!}{(n-r)!\,r!}$ selections

* Properties of Combination

01. Numbers of ways to select 0 items out of N items

$$^{N}C_{0} = \frac{N!}{0!\,(N-0)!} = \frac{\cancel{N!}}{1 * \cancel{N!}} = 1$$

02 No. of ways to select N items out of N items

$$^{N}C_{N} = \frac{N!}{N! * (N-N)!} = \frac{\cancel{N!}}{\cancel{N!} * 0!} = 1$$

03 No. of ways to select r items out of N

$$^{N}C_{r} = \frac{N!}{(N-r)!\,r!}$$

04. No. of ways to select $\underline{N-r}$ items out of N

$$^{N}C_{N-r} = \frac{N!}{(N-(N-r))! * (N-r)!}$$

$$= \frac{N!}{r! * (N-r)!}$$

$$\boxed{^{N}C_{r} = {}^{N}C_{N-r}}$$

* Given n distinct elements, select r items

P_1 P_2 P_3

____ ____ ____ ____ (N^{t} item)

select: $^{N-1}C_{r-1}$

Not select: $^{N-1}C_{r}$

$$\boxed{^{N}C_{r} = {^{N-1}C_{r-1}} + {^{N-1}C_{r}}}$$

P_1 P_2 P_3 P_4

4 players, 3 select

✓: $^{3}C_{2}$

×: $^{3}C_{3}$

* **Pascal Triangle for N = 4** → Generate entire pascal △

1					$^0C_0$				
1	1				$^1C_0$	$^1C_1$			
1	2	1			$^2C_0$	$^2C_1$	$^2C_2$		
1	3	3	1		$^3C_0$	$^3C_1$	$^3C_2$	$^3C_3$	
1	4	6	4	1	$^4C_0$	$^4C_1$	$^4C_2$	$^4C_3$	$^4C_4$

Brute force → For each & every position, generate values through **factorial** & print it

$5! = 120$

$10! = 3628800$

$20! = 2.4 * 10^{18}$

} factorial grows rapidly

	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

* ${}^{N}C_0 = 1$
* ${}^{N}C_N = 1$
* ${}^{N}C_r = {}^{N-1}C_{r-1} + {}^{N-1}C_r$

${}^{4}C_2 = {}^{3}C_1 + {}^{3}C_2$

```
* int [][] res = new int [N+1] [N+1]

    res[0] [0] = 1

for ( i=1 ; i<=n ; i++){
    for ( j=0; j<=i ; j++){
        if (j==0 || j==i)  res[i][j]=1
        else {
            res[i][j]= res [i-1][j-1] + res[i-1][j]
        }
    }
}
```

TC : $O(n^2)$

SC : $O(1)$

Some specific examples where Pascal's triangle is used:

- Google's PageRank algorithm uses Pascal's Triangle to calculate website rankings.
- Medical research uses Pascal's Triangle to model disease spread and population growth.
- Data compression algorithms, like Huffman coding, rely on Pascal's Triangle.

* Excel coloumn title

A	B	C	D	. . . .	Z	AA	AB	AC	. . .	AZ	BA	BB	. . .	ZZ
1	2	3	4		26	27	28	29		52	53	54		

N = 3 → Ans = C

N = 30 → Ans = AD

N = 50 → Ans = AX

N = 52 → Ans = AZ

0	10	20
1	11	
2	12	
3	13	
4	14	
5	.	
6	.	
7	.	
8	.	
9	19	99

A	AA	BA
B	AB	BB
C	AC	BC
D	AD	BD
E	.	.
.	.	.
Z	AZ	BZ

Series with base 26

Map {

1 → A
2 → B
3 → C
4 → D
.
.
.
26 → Z

26 | 30 | 4 D
26 | 1 | 1 A
| 0

⇒ AD

26 | 50 | 24 → X
26 | 1 | 1 → A
| 0

A X

%26 → {0 to 25}

52 = 26 * 2 + 0

52 = 26 * 1 + 26

26 | 52 | ~~0~~ 26 → Z
26 | ~~2~~ 1 | 1 → A
| 0

= A Z

26 | 350 | 12
26 | 13 | 13
| 0

= ML

```
* Create map
  String ans = " ";
  while ( N != 0){
      rem= N % 26
      N = N / 26
      if (rem==0) { rem= 26 , N = N-1 }
      ans= map.get (rem) + ans;
  }
return ans;
```

TC : $O(\log_{26} N)$

SC : $O(1)$

26	600	2 → B
26	23	23 → X
	0	

XB

A → 0

B → 1

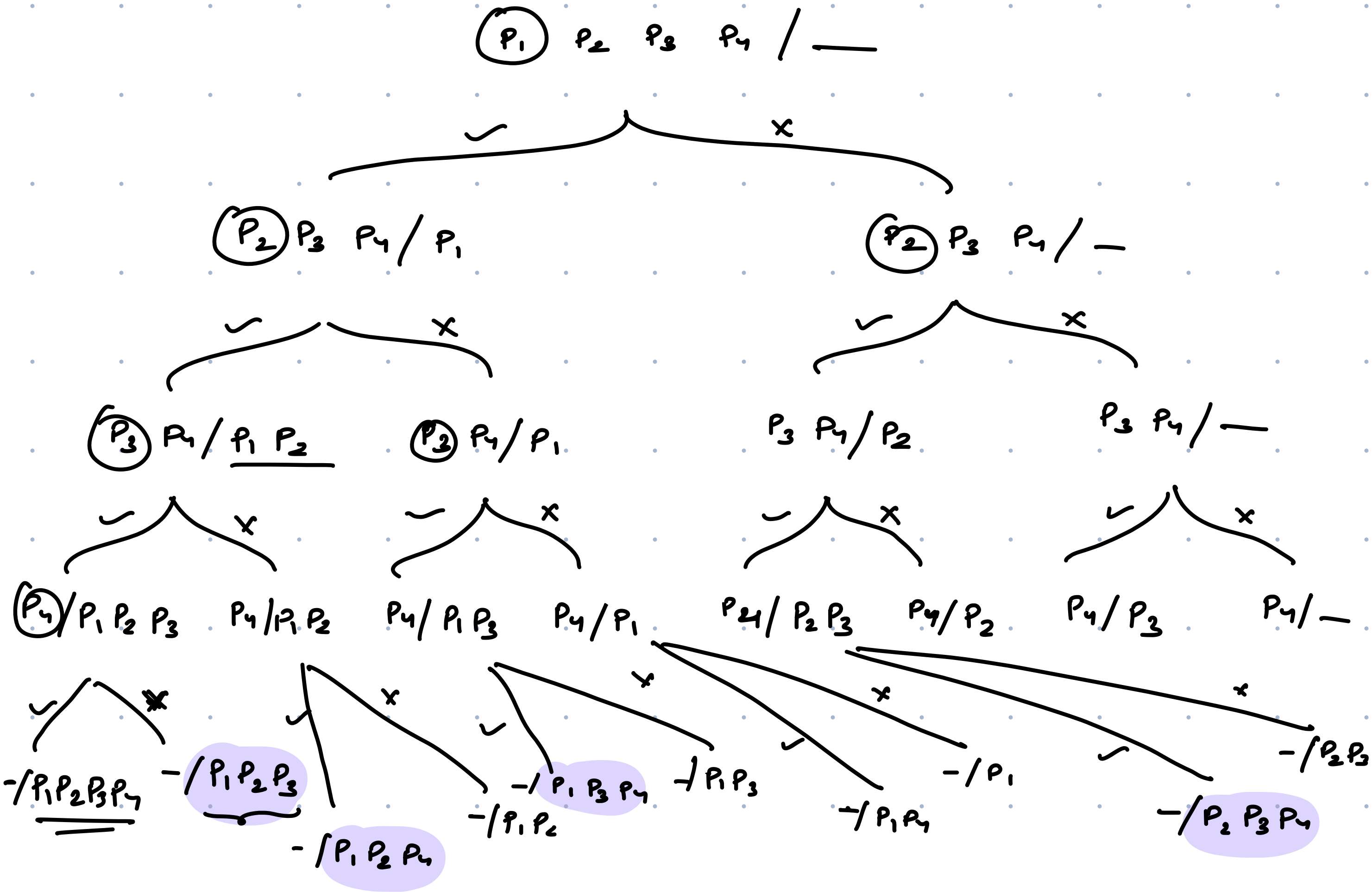
C → 2

⋮

Z → 25

26	52 − 1 = 51	25 → Z
26	1 − 1 = 0	0 → A
	0	

AZ


p1 p2 p3 p4 / —
p2 p3 p4 / p1
p2 p3 p4 / —
p3 p4 / p1 p2
p3 p4 / p1
p3 p4 / p2
p3 p4 / —
p4 / p1 p2 p3
p4 / p1 p2
p4 / p1 p3
p4 / p1
p4 / p2 p3
p4 / p2
p4 / p3
p4 / —
— / p1 p2 p3 p4
— / p1 p2 p3
— / p1 p2 p4
— / p1 p2
— / p1 p3 p4
— / p1 p3
— / p1 p4
— / p1
— / p2 p3 p4
— / p2 p3