Lecture :- Maths : GCD

Agenda

- Permutation
- Combination
- Gcd basics
- Delete one

Permutation

- Arrangement of objects.
- In permutation, order matters
- To simplify $(i, j) \, != (j, i)$

abc : abc
bac
cba
acb
bca
cab

<u>Qu1</u> Given 3 distinct characters. In how many ways can we arrange them ?

s = "abc"

abc bac cab
acb bca cba

$\underline{3} \quad \underline{2} \quad \underline{1} = 3 * 2 * 1 = 3!$

a
b
c

a	b c	c b
b	a c	c a
c	a b	b a

In how many ways n distinct characters can be arranged ?

n!

nPr formulae

Qu. Given 5 distinct characters, in how many places can we arrange them in 2 places?

a b c d e.

ab	ba	ca	da	ea
ac	bc	cb	db	eb
ad	bd	cd	dc	ec
ae	be	ce	de	ed

$\underline{5}\ \underline{4} = 5*4 = 20$

Qu. Given 5 distinct characters, in how many places can we arrange them in 3 places?

$\underline{\ \ }\ \underline{\ \ }\ \underline{\ \ } = 5*4*3 = 60$

5 4 3

Qu. Given 5 distinct characters, in how many places can we arrange them in 4 places?

$5*4*3*2 = 120$

Qu. Given n distinct characters, in how many places can we arrange them in 3 places?

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

Qu. Given n distinct characters, in how many places can we arrange them in 4 places?

$n * n-1 * n-2 * n-3$

Qu. Given n distinct characters, in how many places can we arrange them in r places?

$\underline{n} \quad \underline{n-1} \quad \underline{n-2} \quad \underline{n-3} \quad \underline{n-(r-1)}$

rth position

$Ans = n * n-1 * n-2 * n-3 \text{ - - - - - } n-(r-1)$

Multiply & divide by $(n-r)!$

$$\frac{n * n-1 * n-2 \cdots\cdot n-r+1 * n-r * n-r-1 * n-r-2 \cdots 1}{(n-r)!}$$

$$\boxed{\frac{n!}{(n-r)!} = {}^nP_r}$$

n distinct objects, arrange them at r places

Combination

- No of ways to select something
- Order of selection does not matter
- To simplify $(i, j) = (j, i)$

<u>Qu1</u> Given 4 players, count no. of ways of selecting 3 players.

Rohit Kohli Warner Head

$^4C_3 = 4.$

Rohit Kohli Warner
Rohit Kohli head
Kohli Warner head
Rohit Warner head

$\Rightarrow 4$

Qu No. of ways to arrange 4 players in 3 slots.

Rohit Kohli Conway Gill

1. Rohit
Kohli
Conway.

R	K	C
R	C	K
K	r	C
K	C	r
C	r	K
C	K	r

6 arrangements

2. Ronit
Kohli
Gill

R	K	g
R	g	K
K	r	g
K	g	r
g	r	K
g	K	r

6

3. Kohli
Conway
Gill

⋮

6

4. Ronit
Conway
Gill

⋮

6

for every possible selection = 6 arrangements.

1 sel = 6 arrangements

4 sel = 6 * 4 = 24 arrangements.

Qu Given n distinct items, how many ways can we select r items. & such that $0 <= r <= n$.

arrange

1 selection $= r!$ arrangements.

$r!$ arrangements $= 1$ selection

1 arrangement $= \frac{1}{r!}$ selection.

^nP_r arrangements $= \frac{^nP_r}{r!}$

$$\boxed{^nC_r = \frac{n!}{r!\,(n-r)!}}$$

Property of combination

Property 1 No. of ways of selecting 0 items from n items, i.e, no. of ways to not select anything, will always be 1.

$$^{n}C_{0} = \frac{n!}{0!\,(n-0)!} = \frac{n!}{n!} = 1.$$

Property 2 No. of ways of selecting n items from n items, i.e, no. of ways to not select anything, will always be 1.

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

$$\boxed{^{n}C_{0} = {^{n}C_{n}}}$$

Property 3 No. of ways of selecting $(n-r)$ items from n items, i.e, no. of ways to not select anything, will always be 1.

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

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

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

$$\boxed{^nC_{n-r} = {}^nC_r}$$

Property 4 Given n distinct elements, select r items ^nC_r

n th element

Select Not selected

$$^{n-1}C_{r-1} + {}^{n-1}C_r$$

$$\boxed{^nC_r = {}^{n-1}C_{r-1} + {}^{n-1}C_r}$$

Example: Srk Ranbir Akshay Varun. (3 actors)

s	s	s	r
r	a	r	a
a	v	v	v.

Select: srk $^3C_2$ s r a, s r v, s a v.

Not selecting: $^3C_3 \longrightarrow$ ranbir Akshay varun

(1)

$$\text{Ans} = {}^3C_2 + {}^3C_3$$

$$^{n-1}C_{r-1} + {}^{n-1}C_r.$$

Pascal's triangle

n = 3

1
1 1
1 2 1

0 — 1
1 — 1 1
2 — 1 2 1
3 — 1 3 3 1
4 — 1 4 6 4 1
5 — 1 5 10 10 5 1

0 — 1	${}^{0}C_0$
1 — 1 1	${}^{1}C_0$ ${}^{1}C_1$
2 — 1 2 1	${}^{2}C_0$ ${}^{2}C_1$ ${}^{2}C_2$
3 — 1 3 3 1	${}^{3}C_0$ ${}^{3}C_1$ ${}^{3}C_2$ ${}^{3}C_3$
4 — 1 4 6 4 1	${}^{4}C_0$ ${}^{4}C_1$ ${}^{4}C_2$ ${}^{4}C_3$ ${}^{4}C_4$
5 — 1 5 10 10 5 1	${}^{5}C_0$ ${}^{5}C_1$ ${}^{5}C_2$ ${}^{5}C_3$ ${}^{5}C_4$ ${}^{5}C_5$

Example:

$${}^{5}C_2 = {}^{4}C_1 + {}^{4}C_2.$$

$${}^{n}C_r = {}^{n-1}C_{r-1} + {}^{n-1}C_r$$

solved using DP.

$$\underset{n}{c[i]}\,\underset{r}{[j]} = c[\underset{n-1}{i-1}][\underset{r-1}{j-1}] + c[\underset{n-1}{i-1}][\underset{r}{j}]$$

Pseudo-code

```
int[][] pascalTriangle(int n) {
        c[n][n];
        for(i=0; i<n; i++) {
            c[i][0] = 1;
            c[i][i] = 1;

            for (j=1; j<i; j++) {
                c[i][j] = c[i-1][j-1] + c[i-1][j];
            }
        }
        return c;
}
```

TC: $O(n^2)$
SC: $O(n^2)$ | $O(1)$

Qu: finding nth column title. (Amazon, Google) Excel sheets

1	2	3	.....	25	26	27	28	...	52	53	54	....
A	B	C		Y	z	aa	ab		az	ba	bb	...

n = 30 → ad

n = 50 ax

n = 79 z (26) az (52) bz (78) ca (79.)

n = 78 bz.

0 - a
1 - b
2 - c
3 - d
⋮ u v w
23 - x
24 - y
25 - z.

n = 50

26	50 - 1 = 49	23
26	1 - 1 = 0	0

x
a

ax

n = 99.

26	99 - 1 = 98	20
26	3 - 1 = 2	2

u
c

cu

Pseudocode

0 — a
1 — b
2 — c

```
String columnTile(int n) {
    String ans = "";
    while (n>0) {
        int rem = (n-1) % 26;
        ans = getcode(rem) + ans;
        n = (n-1)/26;
    }
    return reverse(ans);
}
```

n = 50

1st itr =

rem = 49 % 26 = 23

ans = x

$n = \frac{49}{26} = 1$

2nd itr

rem = 0 % 26 = 0

ans = xa

$n = \frac{1-1}{26} = 0$

TC: $\log_{26} n$.

SC: O(1)

Break: 8:40 – 8:50

Gcd basics

Gcd → Greatest common divisor.

hcf → highest common factor

gcd = hcf.

gcd(a, b) → greatest factor dividing a & b

If we have gcd(a, b) = x.

$a \% x = 0$

$b \% x = 0$

x is highest factor of (a, b)

Example

gcd(15, 25) = 5

gcd(12, 30) = 6

gcd(10, -25) = 5

gcd(0, 4) = 0 ✗
4 ✓

$0 \% 4 = 0$
$4 \% 4 = 0$

gcd(0, -10) =

$0 \% 10 = 0$ ✓
$-10 \% 10 = 0$ ✓

= 10

gcd(0, 0) =

0 % 1
% 2
% 3
⋮
∞

0 % 1
% 2
% 3
⋮
∞

infinity

Properties of GCD

$gcd(a, b) = gcd(b, a)$

$gcd(4, 6)$ $gcd(6, 4)$

2   2

$gcd(0, A) = |A|$ abs(A)

$gcd(0, 4)$

$gcd(0, -4)$

$gcd(a, b, c) = gcd(a, gcd(b, c))$

$gcd(gcd(a, b), c)$

$gcd(gcd(a, c), b)$

$gcd(\underbrace{4, 8}_{4}, 18) = \underline{2}$

$gcd(\underbrace{8, 4}_{4}, 18) = 2$

$gcd(\underbrace{8, 18}_{2}, 4) = 2$

Special property of gcd

$gcd(8632, 8650) = gcd(8632, 18)$

tough to find   easy to find

Euclid algorithm

* $gcd(a, b) = \boxed{gcd(a - b, b)}$ $a, b > 0$ & $a >= b$

$$\boxed{gcd(a, b) = gcd(a \% b, b)}$$

Extended Euclid algo.

$gcd(23, 5)$

$\downarrow$

$gcd(3, 5)$

$gcd(23, 5) = 1$

$\downarrow$

$gcd(18, 5)$

$\downarrow$

$gcd(13, 5)$

$\downarrow$

$gcd(8, 5)$

$\downarrow$

$gcd(3, 5) = 1$

Qu Write a function to find gcd(a, b).

a>b
b>a

```
int gcd(a, b) {
    if( b==0) {
        return a;
    }
    return gcd(b, a%b);
}
```

TC: $\log(\max(a, b))$
SC: Recursive space.

gcd(120, 270)
↓
gcd(120, 30)
↓
gcd(0, 30)
↑
ans = 30

gcd(380, 140)
↓
380 % 140
gcd(140, 100)
↓
140 % 100
gcd(100, 40)
↓
100 % 40
gcd(40, 20)
↓
40 % 20
gcd(20, 0)

Qu: Given an array, calculate gcd of entire array.

arr = | 6 | 12 | 15 | $\longrightarrow$ gcd(arr) = 3

```
int gcdArray(arr[]){
        ans = arr[0];
        for(i=1; i<arr.length; i++){
            ans = gcd(ans, arr[i]);
        }
        return ans;
}
```

Qu: Given arr[n] elements. delete one element such that gcd of remaining elements becomes maximum.

	0	1	2	3	4
arr[] =	24	16	18	30	15

Delete one element of array.	gcd of remaining el of array.	ans.
~~24~~, 16, 18, 30, 15	gcd(16, 18, 30, 15)	1
24, ~~16~~, 18, 30, 15	gcd(24, 18, 30, 15)	3
24, 16, ~~18~~, 30, 15	gcd(24, 16, 30, 15)	1
24, 16, 18, ~~30~~, 15	gcd(24, 16, 18, 15)	1
24, 16, 18, 30, ~~15~~	gcd(24, 16, 18, 30)	2

Brute force: $\boxed{n^2 \log max(a, b)}$

Approach

arr[] =

0	1	2	3	4
24	16	18	30	15

prefix gcd[] =

0	1	2	3	4
24	8	2	2	1

suffix gcd[] =

0	1	2	3	4
1	1	3	15	15

After deleting	
0th idx gcd(16, 18, 30, 15)	suffix[i+1] suffix[1]
1st idx gcd(gcd(24), gcd(18, 30, 15))	gcd(pf[i-1], sf[i+1]) gcd(pf[0], sf[2])
2nd idx gcd(24, 16, 30, 15) ⋮	gcd(pf[1], sf[3])

ith idx: gcd(pf[i-1], sfx[i+1])

Edge case:

$i = 0$ sfx[i+1] = sfx[1]

$i = n-1$ pf[i-1] = pf[n-2]

Code: H|W

Thankyou :)