MODULUS & GCD

Good Morning

Content

* Modulus

01. Modular Arithmetic

03. Count pair with sum mod = 0

04. Intro to GCD

05. Properties of GCD

06. Delete one

A % B $\Rightarrow$ Remainder when A is divided by B.

Dividend = div * quo + remainder

$$\text{rem} = \text{Dividend} - \underbrace{\text{div} * \text{quo}}_{\text{largest mul of div} \le \text{Dividend}}$$

01. $10 \% 4 = 2$

$= 10 - (\text{largest mul of } 4 \le 10)$

$= 10 - 8 \Rightarrow 2$

02. $13 \% 5 = 3$

$= 13 - (\text{largest mul of } 5 \le 13)$

$= 13 - 10 = 3$

03. $-40 \% 7 = 2$

$= -40 - (\text{largest mul of } 7 \le -40)$

$= -40 - (-42)$

$= -40 + 42 = 2$

04. $-60 \% 9 = 3$

$= -60 - (\text{largest mul of } 9 \le -60)$

$= -60 - (-63)$

$= -60 + 63 = 3$

	min Ans	max Ans
$x \% M$	0	$M-1$

Why % M ?

→ Restrict the output in given range.

$$\left.\begin{matrix} -\infty \\ \vdots \\ \infty \end{matrix}\right\} \% M \Rightarrow \{0 \text{ to } (M-1)\}$$

$M = 10^9 + 7$ → prime no. → why prime ? DSA 4.2

↳ very close to int range

* Modular Arithmetic → % with (+, −, *, /)

01. $(a+b)\%m \Rightarrow (a\%m + b\%m)\%m$

$a = 13$

$b = 9$

$m = 5$

$(13+9)\%5 = (13\%5 + 9\%5)\%5$

$22\%5 = (3+4)\%5$

$= 2 \qquad = 7\%5$

$\Rightarrow 2 \qquad = 2$

02. $(a * b) \% m \Rightarrow (a \% m * b \% m) \% m$

$a = 13$

$b = 9$

$m = 5$

$(13 * 9) \% 5 = (13 \% 5 * 9 \% 5) \% 5$

$\Rightarrow 117 \% 5 = (3 * 4) \% 5$

$= 2 = 12 \% 5$

$\Rightarrow 2 = 2$

<u>03</u> $(a - b) \% m = (a \% m - b \% m) \% m$

$a = 13$

$b = 9$

$m = 5$

$(13 - 9) \% 5 = (13 \% 5 - 9 \% 5) \% 5$ ×

$4 \% 5 = (3 - 4) \% 5$

$\Rightarrow 4 = -1 \% 5 = \underline{-1}$ 4 in python

$-1 + 5$ -1 in Java, C++, C#

$= \underline{4}$

03. $(a - b) \% m = (a \% m - b \% m + m) \% m$

$= (13 \% 5 - 9 \% 5 + 5) \% 5$

$= (3 - 4 + 5) \% 5$

$= 4 \% 5 \Rightarrow \underline{4}$

Fermat theorem

$\Uparrow$

04. $(a/b)\%m \Rightarrow \boxed{(a * b^{-1})\%m} \longrightarrow$ Inverse modulo

$\Downarrow$

DSA 4.2

05. $a\%m = (((a\%m)\%m)\%m)$

06. $a^b\%m = ((a\%m)^b\%m)$

* **Quiz**

$(37^{103} - 1)\%12$

$\Rightarrow (37^{103}\%12 - 1\%12 + 12)\%12$

$= ((37\%12)^{103}\%12) - 1 + 12)\%12$

$= ((1)^{103}\%12) - 1 + 12)\%12$

$\Rightarrow (1 - 1 + 12)\%12$

$= 12\%12 \Rightarrow \underline{0}$

Q Given A[ ], m. Calculate no. of pairs (i, j) such that

(A[i] + A[j]) % m = 0

Note :- i != j && pair (i, j) = pair (j, i)

$$A[\,] = \{\underset{0}{4}\ \ \underset{1}{7}\ \ \underset{2}{6}\ \ \underset{3}{5}\ \ \underset{4}{5}\ \ \underset{5}{3}\} \qquad m = 3$$

i	j	A[i]	A[j]	(A[i] + A[j]) % m

0	3	4	5	$(4+5)\%3 = 0$
0	4	4	5	$(4+5)\%3 = 0$
1	3	7	5	$(7+5)\%3 = 0$
1	4	7	5	$(7+5)\%3 = 0$
2	5	6	3	$(6+3)\%3 = 0$

Ans = 5

* Brute force → Check for all possible pair

if $(A[i] + A[j])\%m == 0)$ ans ++;

TC : $O(n^2)$

SC : $O(1)$

* Idea 2

$$(A[i] + A[j])\ \%\ m = 0$$

$$(A[i]\%m + A[j]\%m)\ \%\ m = 0$$

$0 + 0$

$1 + (m-1)$

$2 + (m-2)$

$3 + (m-3)$

$\vdots \quad \vdots \quad \vdots$

$k + (m-k)$

A[ ] = { 13 14 22 3 32 19 16 } m = 4

rem[ ] = { 1 2 2 3 0 3 0 }

Ans = 4

m = 5

A[ ] = { 6 7 5 11 19 20 9 15 14 13 12 23 }

6	7	5	11	19	20	9	15	14	13	12	23
0	1	2	3	4	5	6	7	8	9	10	11

rem[ ] = { 1 2 0 1 4 0 4 0 4 3 2 3 }

1	2	0	1	4	0	4	0	4	3	2	3
0	1	2	3	4	5	6	7	8	9	10	11

range of rem = { 3 2 2 2 3 }

3	2	2	2	3
0	1	2	3	4

Pair of rem

1 & 4 = $2 * 3 = 6$

2 & 3 = $2 * 2 = 4$

0 & 0 = $\underbrace{{}^{3}C_{2}}_{\frac{n*(n-1)}{2}} = 3$

Ans = 13

$m = 10$

ar[] =

29	11	21	17	2	5	4	6	23	13	26	14	18	15	30	35	50	20	40	9
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

rarr[] =

9	1	1	7	2	5	4	6	3	3	6	4	8	5	0	5	0	0	0	9

$m = 10$

cnt[10] =

4	2	1	2	2	3	2	1	1	2
0	1	2	3	4	5	6	7	8	9

A[i] % m   A[j] % m

cnt[0] $\longrightarrow$ ${}^4C_2$ pairs

cnt[1] * cnt[9] $\longrightarrow$ 2 * 2 = 4

cnt[2] * cnt[8] $\longrightarrow$ 1 * 1 = 1

cnt[3] * cnt[7] $\longrightarrow$ 2 * 1 = 2

cnt[4] * cnt[6] $\longrightarrow$ 2 * 2 = 4

cnt[5] $\longrightarrow$ ${}^3C_2$ pairs

```
int countpairs (A[], m)
    int [] cnt = new int [m];

    for (i=0; i<n; i++){
        int rem = A[i] % m
        cnt [rem]++;
```

TC : O(n+m)

SC : O(m)

ans = 0

int x = cnt [0]

ans = ans + $\frac{x * (x-1)}{2}$;

if (m % 2 == 0){

int y = cnt [m/2];

ans = ans + $\frac{y * (y-1)}{2}$

}

only when m = even

// get all remaining pairs

i = 1 , j = m - i

while (i < j)

ans = ans + cnt [i] * cnt [j]

i++;

j--;

}

return ans;

}

8 : 38 AM

GCD ⇒ Greatest common Divisor

HCF = Highest common factor

GCD(A, B) ⇒ largest no. that divides both A & B.

$GCD(A, B) = x$ - $\left.\begin{array}{l} A \% x = 0 \\ B \% x = 0 \end{array}\right\}$ x is largest common factor

GCD(15, 25)

15	25
1	1
3	5
5	25
15	

Ans = 5

GCD(12, 30)

12	30
1	1
2	2
3	3
4	5
6	6
12	10
	15
	30

Ans = 6

GCD(10, -25)

10	-25
1	-25
2	-5
5	-1
10	1
	5
	25

Ans = 5

GCD(0, 4)

0	4
1	1
2	2
3	4
4	
⋮	
∞	

Ans = 4

GCD(0, -4)

0	-4
1	-4
2	-2
3	-1
4	1
5	2
⋮	4
∞	

Ans = 4

Properties of GCD

01. $GCD(A, B) = GCD(B, A)$

02. $GCD(A, B) = GCD(|A|, |B|)$

03. $GCD(0, A) = |A|$

04. $GCD(A, B, C) = GCD(A, GCD(B, C))$

$= GCD(B, GCD(A, C))$

$= GCD(C, GCD(A, B))$

* special property of GCD (true for $A \geq B$)

$GCD(A, B) = x \longrightarrow A \% x = 0$

$B \% x = 0$

$GCD(A - B, B) = x$

$(A - B) \% x = 0$ $\qquad B \% x = 0$

$\Downarrow$

$(\underbrace{A \% x}_{} - \underbrace{B \% x}_{} + x) \% x$

$(0 - 0 + x) \% x$

$x \% x = \underline{\underline{0}}$

$GCD(A, B) = GCD(A - B, B)$

* Calculate $GCD(23, 5)$

$$GCD(23,5) = GCD(\overset{23-5}{18},5) = GCD(\overset{18-5}{13},5) = GCD(\overset{13-5}{8},5) = GCD(\overset{8-5}{3},5)$$

$$GCD(23,5) = GCD(3,5)$$
$$= GCD(23\%5, 5)$$

$$GCD(A,B) = GCD(A-1B, B)$$
$$= GCD(A-2B, B)$$
$$= GCD(A-3B, B)$$
$$\vdots$$
$$= GCD(\underbrace{A-xB}, B)$$

A - greatest value of $B \le A$

$$GCD(A,B) = GCD(A\%B, B)$$

* write a function for $GCD(A,B)$

01. $GCD(24, 16) = GCD(8,16) = GCD(8,16) = GCD(8,16)$ → Infinite loop

$$GCD(A,B) = GCD(B, A\%B)$$

Q2 GCD(24, 16) = GCD(16, 8) = GCD(8, 0) ⇒ **Ans = 8**

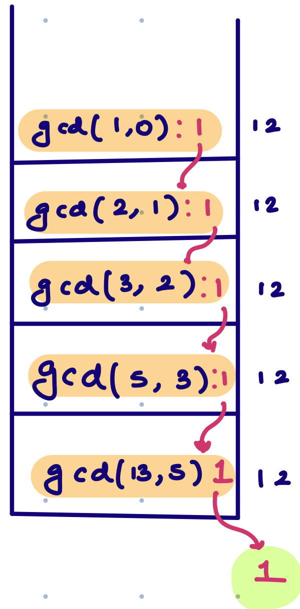
Q3. GCD(14, 21) = GCD(21, 14) = GCD(14, 7) = GCD(7, 0) **Ans = 7**

Q4. GCD(13, 9) = GCD(9, 4) = GCD(4, 1) = GCD(1, 0) **Ans = 1**

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

TC : O(log max(a,b))

SC : O(log max(a,b))



* Given an A[], calculate GCD of entire array.

A[3] = {6, 12, 15}

ans = 0 → 6 → 6 → 3

Ans = 3

```
int gcdArr(int [] A){
    int ans = 0
    for (i=0; i<n; i++){
        ans = gcd(ans, A[i]);
    }
    return ans;
}
```

TC : O(n * log max(arr))

* Delete One

Given A[N] elements, we have to delete one element such that GCD of all remaining elements become maximum.

A[ ] = {24 16 18 30 15}

	GCD
{~~24~~ 16 18 30 15}	1
{24 ~~16~~ 18 30 15}	3 ← Ans
{24 16 ~~18~~ 30 15}	1
{24 16 18 ~~30~~ 15}	1
{24 16 18 30 ~~15~~}	2

Brute force → Delete an ele & calculate the GCD for rest of the array

Repeat this process N times

TC: $O(N^2 \log \max(arr))$

pfgcd [i] = gcd of all elements from 0 to i

sfgcd [i] = gcd of all elements i to n-1

Idea Use prefix Arrays

```
int deleteone (int [] A)
    int [] pfgcd = pf (A);
    int [] sfgcd = sf (A);
    ans = 0
    for ( i=0 ; i<n; i++){
        // delete A[i]
        A[] = {A0 A1 A2 A3 .... Ai-1 Ai Ai+1 ... An-1}
        // left = gcd of all ele 0 to i-1
        int left = 0
        if (i>0) left = pfgcd [i-1];
        // right = gcd of all ele from i+1 to n-1
        int right = 0
        if (i < n-1) { right = sfgcd [i+1] }
        int val = gcd(left, right);
        ans = max(ans, val);
    }
```

```
}

int [] pf ( int [] A ){

    int [] pfgcd = new int[A.length]
     ans = 0
    for ( i=0; i<n; i++){
        ans = gcd (ans, A[i]);
        pfgcd[i] = ans;
    }
    return pfgcd;
}
int [] sf ( int [] A ){

    int [] sfgcd = new int[A.length]
     ans = 0
    for ( i=n-1 ; i≥0 ; i-- ){
        ans = gcd (ans, A[i]);
        sfgcd[i] = ans;
    }
    return sfgcd;
}
```