Modular Arithmetic and GCD

Today's Content:

- Modular Arithmetic
- Count pairs whose sum % m = 0
- Intro to GCD
- Properties of GCD
- Delete one

- Maths
- GCD
- Combinotrics
- Modular Arithmetic

Modular Arithmetic

A % B → Remainder when A is divided by B.

↳ [0, B-1]

x % 6 → [0, 5]

30 % 7 → 2 ⇒ 30 - 7 - 7 - 7 - 7 (4 * 7 = 28)

40 % 9 → 4

5 % 5 → 0

1 % 2 → 1

A % B → keep subtracting B from A till A < B.

(or) A % B → Dividend - Greatest multiple of divisor <= dividend.

-3 % 7 = -3 + 7 (JAVA or C++)

30 % 7 = 30 - 7 * 4 = 2

-7 % 3 = -7 - (3 * -3) = -7 - (-9) = 2

-30 % 7 = -30 - (7 * -5) = -30 - (-35) = 5

Rules of Modular Arithmetic

1) $(a+b)\%m = (a\%m + b\%m)\%m$.

$[0, m-1]$ $\quad [0, m-1] + [0, m-1]$

$= [0, 2m-2]\%m$

$= [0, m-1]$

$a = 9 \quad b = 8 \quad m = 5$.

$(9+8)\%5 = (9\%5 + 8\%5)\%5$

$17\%5 = (4+3)\%5$

$2 = 7\%5$

$\boxed{2 = 2}$

2) $(a * b)\%m = ((a\%m) * (b\%m))\%m$

3) $(a+m)\%m = (a\%m + m\%m)\%m$

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

$= a\%m$.

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

$([0, m-1] - [0, m-1]) < 0$

$(7-10)\%5 = -3\%5 = (-3+5)\%5 = 2\%5 = 2$

5) $(a^b) \% m = (a*a*a* \ldots \text{b times}) \% m$

$= ((a \% m)*(a \% m)*(a \% m) \ldots) \% m$

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

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

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

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

$= ((1^{103}) \% 12 + 11) \% 12$

$= (1 + 11) \% 12$

$= 0$

Quiz 2: $(25 + 13) \% 7 = 38 \% 7 = \boxed{3}$

$(25 \% 7 + 13 \% 7) \% 7$

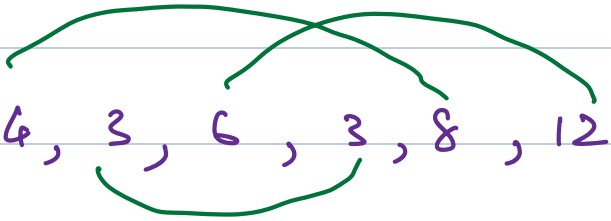
$(4 + 6) \% 7$

$10 \% 7$

$= \boxed{3}$

Question: Given N array elements, count pairs (i, j) such that (arr[i] + arr[j]) % m = 0

i != j



ex: arr = {4, 3, 6, 3, 8, 12} m = 6

O/P = 3 (If duplicate pairs, then 6)

Brute force approach

- Iterate with two loops through entire array.
- Calculate sum % m
- If result == 0, cnt++.

```
for(i=0; i<N; i++){
    for(j=i+1; j<N, j++){
        if((arr[i] + arr[j])%m == 0) cnt++;
    }
}
```

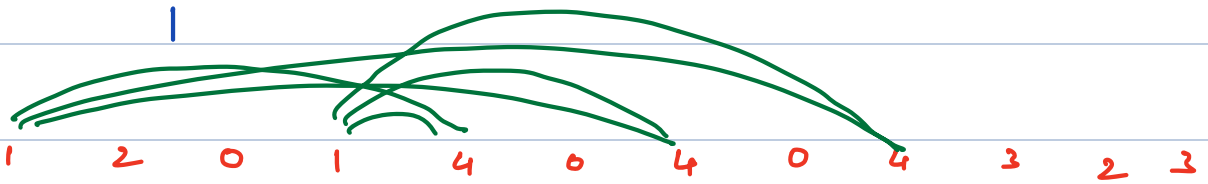
T.C = O(N²) S.C = O(1).

$(a+b)\%m = 0$

$(a\%m + b\%m)\%m = 0$

$[0, m-1]$ $[0, m-1]$

0	0
$1 \to i$	$m-1$
2	$m-2$
⋮	
$m-1$	1



arr = {6, 7, 5, 11, 19, 20, 9, 15, 14, 13, 12, 23}
0 1 2 3 4 5 6 7 8 9 10 11

$m = 5$

1) Take arr[i]%m and store freq in a hash map.

0 → ✗ ✗ 3

1 → ✗ 2

2 → ✗ 2

3 → ✗ 2

4 → ✗ ✗ 3

a%5 a%5

(1, 4) → (0, 4), (0, 6), (0, 8)
(3, 4), (3, 6), (3, 8)

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

0 → 3	0 → 3	= 3
1 → 2	4 → 3	= 6
2 → 2	3 → 2	= 4
3 → 2	2 → 2	= 4
4 → 3	1 → 2	= 6
		23

$\not{3}\ 2$

$\frac{3 * 2}{2} = \frac{6}{2}$

m → Even = 4 →

0	0
1	3
2 → 3	2 → 3

$\not{3}\ 2$

$\frac{3 * \not{2}}{\not{2}} = 3$

$\frac{x * (x-1)}{2}$

Psuedocode

```
func freq_map (int[] arr, int m){
    hm = {}
    for(i = 0 ; i<N ; i++){
        if (hm.get(arr[i]%m))
            hm[arr[i]%m]++;
        else
            hm[arr[i]%m] = 1;
    }
    return hm; }
```

```
func solve (int[] arr, int m){          (i != j)
    hm = freq-map (arr, m);
    cnt = 0
    for (i=0; i<m; i++){
        if (i==0 or (m%2==0 and i==m/2)){
            cnt += hm[i] * (hm[i] - 1)/2
        }
        else
            cnt += hm[i] * hm[m-i];
    } return cnt;
}
```

```
func solve (int[] arr, int m){          (i < j)
    hm = freq-map (arr, m);
    cnt = 0
    for (i=0; i<=m/2; i++){
        if (i==0 or (m%2==0 and i==m/2)){
            cnt += hm[i] * (hm[i] - 1)/2
        }
        else
            cnt += hm[i] * hm[m-i];
    } return cnt; }
```

$TC = O(n+m)$

$a \% m \rightarrow [0, m-1] \rightarrow$ freq. map.

$S.C = O(m)$

GCD / HCF → Highest Common Factor.

↳ Greatest Common Divisor

$gcd(a, b) = x \rightarrow$ x is a factor of a
x is a factor of b.
x is max of all common factors

$gcd(15, 25) = 5$

15	25
1	1
3	5
5	25
15	

$gcd(12, 30) = 6$

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

$gcd(0, 4) = 4$

0	4
$-\infty$	1
⋮	2
⋮	4
$+\infty$	

$gcd(0, 0) =$ Undefined.

0	0
$-\infty$	$-\infty$
⋮	⋮
⋮	⋮
∞	∞

gcd (4, 7) = 1

1 1

2 7

4

Properties of GCD

1) gcd (a, b) = gcd (b, a)
2) gcd (0, x) = x
3) gcd (a, b, c) = gcd (a, gcd (b, c))
= gcd (b, gcd (a, c))
= gcd (c, gcd (a, b))

4) gcd (a, b) = gcd (a-b, b) = gcd (b, a-b)

5) gcd (a, b) = gcd (a%b, b) = gcd (b, a%b)

Quiz 6

gcd (a1, a2, ... an)

gcd (a1, gcd (a2, gcd (..., gcd (an-1, an))

log (max (an-1, an))

= N * log (max (a1 ... an))

Quiz 7

$gcd(15, 21, 33, 45) = gcd(3, 33, 45)$

$= 3$

15	21
↓	↓
1	1
3	3
5	7
15	21

Psuedocode (gcd(a, b))

```
func gcd (a, b) {
    temp = max(a, b)
     b = min(a, b)   // Min value
     a = temp        // Max value.
    if (b == 0) return a;
    return gcd(b, a % b)
}
```

Question: Given an array, calculate GCD of all the elements in the array.

```
gcd_val = 0
for (i = 0; i < N; i++)
    gcd_val = gcd(gcd_val, arr[i]);
return gcd_val
```

Question: Given an array, delete one element such that gcd of remaining elements is maximum. Return max GCD after deleting.

arr = {24, 16, 18, 30, 15}

arr = {24, 16, 18, 30, 15} = 1

arr = {24, 16, 18, 30, 15} = 3 → O/P.

arr = {24, 16, 18, 30, 15} = 1

arr = {24, 16, 18, 30, 15} = 1

arr = {24, 16, 18, 30, 15} = 2

Quiz 8: arr = {21, 7, 2, 14}

Brute force approach

- Iterate over all elements.
- Remove current element and calculate GCD.
- Return max GCD.

```
max_gcd = 0
for (i = 0 ; i < N ; i++){
    dupl_arr = copy (arr)
    delete arr[i] from dupl_arr
    gcd = 0
    for (j = 0 ; j < N-1 ; j++){
        gcd = gcd (gcd , dupl_arr[j]);
    }
    max_gcd = max (max_gcd , gcd);
}
```

$T.C = N^2 \log(\max(arr))$

Optimal Approach

arr = {24, 16, 18, 30, 15} = 1

PFGCD = {24, 8, 2, 2, 1}

SFGCD = {1, 1, 3, 15, 15}

Psuedocode

```
func solve (int[] arr){
    PFGCD[N] // TODO
    SFGCD[N] // TODO.
    max_gcd = 0
    for(i=0; i<N; i++){.
        if(i==0): cur_pf_gcd = 0
        else: cur_pf_gcd = PFGCD[i-1]
        if(i==N-1): cur_sf_gcd = 0
        else: cur_sf_gcd = SFGCD[i+1]
        cur_gcd = gcd(cur_pf_gcd, cur_sf_gcd)
        max_gcd = max(max_gcd, cur_gcd)
    }
    return max_gcd. }
```

T.C = N log(max(arr))

S.C = N