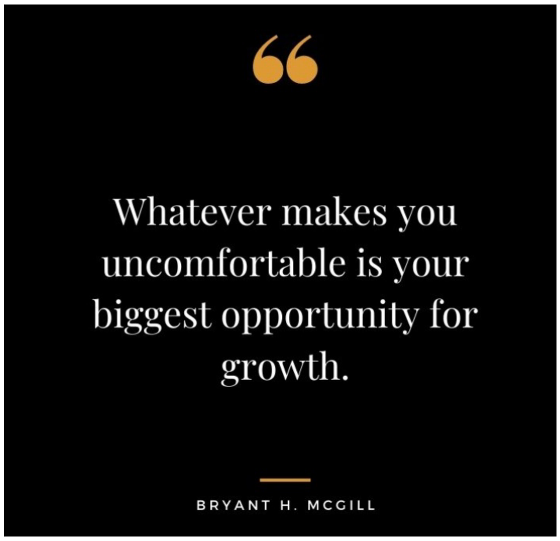

Today's content

- → GCD intro
- → properties of G.C.D
- → GCD function
- → GCD problems
  - → check subsequence with gcd=1
  - → Delete One.

G.C.D : Greatest Common Divisor or Highest Common Factor (H.C.F)

gcd(a,b) : greatest factor which divides both a & b.

↳ x ⇒ $a \% x = 0$ and $b \% x = 0$, x highest factor

$gcd(15, 25) = 5$

15	25
1	1
3	5
5	12
15	

$gcd(12, 30) = 6$

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

$gcd(10, -25) = 5$

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

-ve nos.

$gcd(0, 8) = 8$

0	8
1	1
2	2
3	4
4	8
⋮	
8	
⋮	
∞	

$0 \% 1 = 0$

$0 \% 2 = 0$

$0 \% 8 = 0$

$0 \% 1000 = 0$

$0 \% x = 0$

$gcd(0, -10) = 10$

0	-10
1	1
2	2
3	5
4	10
5	
⋮	
10	
⋮	
∞	

$gcd(-16, -24) = 8$

-16	-24
1	1
2	2
4	3
8	4
16	6
	8
	12
	24

$gcd(-2, -3) = 1$

-2	-3
1	1
2	3

Properties of gcd

→ $\gcd(a,b) = \gcd(b,a)$ {Commutative}

→ $\gcd(a,b) = \gcd(|a|,|b|)$

→ $\gcd(0,x) = |x|$

{Associative}

→ $[\gcd(a,b,c) = \gcd(\gcd(a,b), c) = \gcd(a, \gcd(b,c))]$

Special Property:

$A, B > 0$ and $A > B$

Let's say, $[\gcd(A,B) = x.] \Rightarrow A\%x == 0$, $B\%x == 0$

$$\boxed{\gcd(A-B, B) = x}$$

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

$B\%x == 0$ ✓

$(A\%x - B\%x + x)\%x$

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

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

Eg:

$gcd(23,5) \rightarrow gcd(18,5) \rightarrow gcd(13,5) \rightarrow gcd(8,5) \rightarrow gcd(3,5)$

$\{ gcd(23\%5, 5) = gcd(3,5) \}$

Given $A, B > 0$ and $A > B$.

$$
\begin{aligned}
gcd(A, B) &= gcd(B, A-B) \\
&= gcd(B, A-2B) \\
&= gcd(B, A-3B) \\
&\vdots \\
&\vdots\ gcd(B, \underset{\text{dvd}}{A} - \underset{\text{quo}}{x}\, \underset{\text{div}}{B})
\end{aligned}
$$

{Max no. of times you can subtract B from A.}

$$\boxed{gcd(A,B) = gcd(B, A\%B)}$$

```
int gcd ( int a , int b){
    // if (a==0) return b   {not required}
    if (b==0) return a
    return gcd(B, A%B)
}
```

$gcd(\overset{A}{15}, \overset{B}{6})$

$\downarrow$

return $gcd(6, 3)$ 15%6

$\downarrow$

return $gcd(3, 0)$ 6%3

what if A < B ?

A B
gcd(6, 21) → 3

↓ 6%21

return gcd (21 , 6) → 3

↓ 21%6

return gcd (6 , 3) → 3

↓ 6%3

return gcd (3 , 0) → 3

A B
gcd(7, 0) : return 7.

A B
gcd (0 , 8) → 8

↓

return gcd(8, 0) → 8

A>B.

~~X Wrong.~~

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

Correct ✓

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

A B
gcd(15 , 6)
↓
A B
gcd(3 , 6)
↓
A B
gcd(3 , 6)
↓
A B
gcd(3 , 6)
↓
⋮

→ infinite recursion
→ Stack overflow.

⇒ // gcd(|a|, |b|) → to avoid negative cases.

$$\boxed{T.C \rightarrow O(\log_2 \max(A, B))}$$

Eg.: $N \rightarrow N-1 \rightarrow N-2 \rightarrow \dots \rightarrow 2 \rightarrow 1$: N iterations.

Eg: $N \rightarrow N/2 \rightarrow N/4 \rightarrow N/8 \dots 1$: $\log_2 N$ iterations

$a > b$.

Pivot code: $\gcd(a, b) = \gcd(\underbrace{a \% b}_{< a/2}, b)$

$\{b < \frac{a}{2}\}$

$a \% b < b < \frac{a}{2}$

$$\boxed{a \% b < \frac{a}{2}}$$

$\{b = \frac{a}{2}\}$

$a \% b < b$

$$\boxed{a \% b < \frac{a}{2}}$$

$\{b > \frac{a}{2}\}$

$\Rightarrow 2b > a$

$\Rightarrow 2b - a > 0$

multiply by -1 on both sides

$$\boxed{a - 2b < 0}$$

//add a on both sides

$a - 2b + a < a$

$2a - 2b < a$

$(a - b) < \frac{a}{2}$

$$\boxed{a \% b < \frac{a}{2}}$$

Case 3: {How many times we are subtracting b}

$\boxed{a \% b = a - b}$ {only subtract b once}

$= \cancel{a - 2b}$ ×

Note: — Whenever you multiply both sides by -1, reverse the inequality.

$3 > 1 \Rightarrow -3 < -1$

Q Given N array elements. Calculate GCD of entire array.

arr[3] : [6 12 15] {ans = 3}

0 1 2

6

3

arr[4] : [8 16 12 10] {ans = 2}

0 1 2 3

8

4

2

<u>pseudo-code -</u>

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

T.C → $O(N * \log_2 \text{max element in array})$

⇓

gcd of entire array → {T.C → O(N)} [#todo]

Hint 1. → For every gcd, we are using same variable ans.

Hint 2. → ans. ─┬→ either remains same
            or
         └→ reduce.

Subsequence Basics

A sequence generated by deleting >=0 elements from array.

arr[8]: [3 2 1 6 4 8 10 9]
0 1 2 3 4 5 6 7

1 → [3 2 6 10 9] : YES.

2 → [2 6 10 9 3] : No. {order matters}

arr[4] : [6 9 0 8]
0 1 2 3

1 → [9 8] : YES

2 → [6 9 8] : YES

3 → [0 9 8] : No {order matters}

4 → [6 9 0 8] : YES.

Q) Given an array return if there exists a subsequence with gcd = 1.

{can't consider empty subsequence}

Eg1: arr[5] : [4 6 3 8] → {4, 3, 8} YES.

Eg2: arr[5] : [16 10 6 15 27] → {16, 10, 27} : YES.

Eg3: arr[4] : [6 12 3 18] : No

idea-1. → Consider all the subsequences & get gcd.

T.C → (2^N * T.C to get gcd of a subsequence)

idea-2. A B C D E F

[Subsequence with gcd = 1] Given

gcd(B, C, F) = 1

gcd of entire array

$$= \gcd(A, B, C, D, E, F) = \gcd(\gcd(A, D, E), \gcd(B, C, F))$$

$$= \gcd(\gcd(A, D, E), 1)$$

$$= 1$$

Final observation

[If G.C.D of entire array is 1 ⇒ we have a subsequence with gcd = 1.]

Delete One

Given N array elements, we have to delete 1 element, such that gcd of remaining array is maximum.

$arr[\,] \to [\cancel{\underset{0}{24}}\ \ \underbrace{\underset{1}{16}\ \ \underset{2}{18}\ \ \underset{3}{30}\ \ \underset{4}{15}}_{gcd(\ \ )}]$: 1

$arr[\,] \to [\underset{0}{24}\ \ \cancel{\underset{1}{16}}\ \ \underset{2}{18}\ \ \underset{3}{30}\ \ \underset{4}{15}]$: $\boxed{3}$ {ans}

gcd

$arr[\,] \to [\underset{0}{24}\ \ \underset{1}{16}\ \ \cancel{\underset{2}{18}}\ \ \underset{3}{30}\ \ \underset{4}{15}]$: 1

gcd

$arr[\,] \to [\underset{0}{24}\ \ \underset{1}{16}\ \ \underset{2}{18}\ \ \cancel{\underset{3}{30}}\ \ \underset{4}{15}]$: 1

gcd

$arr[\,] \to [\underbrace{\underset{0}{24}\ \ \underset{1}{16}\ \ \underset{2}{18}\ \ \underset{3}{30}}_{gcd}\ \ \cancel{\underset{4}{15}}]$: 2

idea-1 : Remove every element & get gcd of entire array.

T.C → O(N * T.C for gcd of entire array)

→ O(N^2)

while calculating gcd of entire array, skip the element.

// Assume N = 7.

0 1 2 3 4 5 6

Delete

0 gcd [1-6]

1 gcd (gcd [0-0], gcd [2-6])

2 gcd (gcd [0-1], gcd [3-6])

3 gcd (gcd [0-2], gcd [4-6])

4 gcd (gcd [0-3], gcd [5-6])

5 gcd (gcd [0-4], gcd [6-6])

6 gcd [0-5]

ith. gcd (gcd (0, i-1) , gcd (i+1, N-1))

idea ⇒ pgcd[i] → gcd of all elements [0-i]

sgcd[i] → gcd of all elements [i-N-1]

pseudo-code

```
// construct pgcd[];
    pgcd[0] = arr[0]
    for ( i = 1 ; i < N ; i++ ) {
        pgcd[i] = gcd ( pgcd[i-1], arr[i] );
    }

// construct sgcd[]
    sgcd[N-1] = arr[N-1]
    for( i = N-2 ; i >= 0 ; i-- ) {
        sgcd[i] = gcd (sgcd[i+1], arr[i]);
    }

// try deleting all the elements.
    ans = 0.
    for( i = 0 ; i < N ; i++ ) {
            //delete ith.
        //left -> pgcd[i-1]     i == 0 {edge case}
        //right -> sgcd(i+1]    i == N-1 {edge case}
                                {#todo} handle them.
        ans = max ( ans, gcd(left, right))
    }
return ans;
```

$$\boxed{T.C \rightarrow O(N * \log_2 max)}$$

idea ⇒ pgcd[i] → gcd of all elements [0 - i]
sgcd[i] → gcd of all elements [i - N-1]

arr[ ] → [24 16 18 30 15]
0 1 2 3 4

pgcd[ ] → [24 8 2 2 1]

sgcd[ ] → [1 1 3 15 15]

delete

0 gcd[1-4] = sgcd[1] = 1

1 gcd(gcd[0-0], gcd[2-4]) = gcd(pgcd[0], sgcd[2]) = 3

2 gcd(gcd(0-1), gcd(3-4)) = gcd(pgcd[1], sgcd[3]) = 1

3. gcd(gcd(0-2), gcd(4-4)) = gcd(pgcd[2], sgcd[4]) = 1

4 gcd(0,3) = pgcd[3] = 2