Todays Content: Conten starts from 9:05 pm

- → GCD Intro
- → Properties of GCD
- → GCD function
- → GCD Problems
    - a) Check Subsequence with gcd = 1
    - b) Delete 1 element such that gcd of remaining elements in your array is max

GCD: greatest common divisor / HCF → highest common factor

gcd(a, b) = x { x is greatest number $a \% x == b \% x = 0$ }

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(10, -25) : 5

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

gcd(0, 8) : 8

0	8
1	1
2	2
3	4
4	:
:	8
8	
:	
∞	

gcd(0, -10) : 10

0	-10
1	1
2	2
:	5
10	10
:	
∞	

gcd(-16, -24) : 8

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

gcd(-2, -3)

-2	-3
1	1
2	3

gcd(0, 0) : not defined

↳ If at all, they ever give this input, they will specify what to return

0	0
1	1
2	2
3	3
:	:
∞	∞

Note: $x \% 0$ → { not defined }

$0 \% x = 0$ { defined }

Properties of gcd(a, b):

→ $\gcd(a, b) = \gcd(b, a)$ ✓ // Commutative property

$|x| =$ abs of x

→ $\gcd(a, b) = \gcd(|a|, |b|)$, // sign of number not matter

→ $\gcd(0, x) = |x|$, if $x != 0$ // $\gcd(0, -10) = 10$,

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

associative property

Special Property: $[A, B > 0$ && $A >= B$ && $\gcd(A, B) = x]$

Ass: $\gcd(A, B) = x$: x is greatest number $A \% x = 0$, $B \% x = 0$

→ $\gcd(A - B, B) = x$

if $\underbrace{(A - B) \% x == 0}_{\text{// satisfied ✓}}$ && $\underbrace{B \% x == 0}_{\text{✓ satisfied, given}}$

$= [\underbrace{A \% x}_{0} - \underbrace{B \% x}_{0} + x] \% x$

$= [0 - 0 + x] \% x$

$= [x] \% x = 0$

Claim: if $A, B > 0$ && $A >= B$

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

$$\gcd(\overset{a}{23}, \overset{b}{5}) = \boxed{\overset{23-5}{\gcd(\overset{a}{18}, \overset{b}{5})}} = \boxed{\overset{23-5*2}{\gcd(\overset{a}{13}, \overset{b}{5})}} = \boxed{\overset{23-5*3}{\gcd(\overset{a}{8}, \overset{b}{5})}} = \boxed{\overset{23-5*4}{\gcd(\overset{a}{3}, \overset{b}{5})}}$$

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

$$\boxed{A\%B = A - \text{greatest mul of } B \propto = A}$$

$A >= B > 0$

$\gcd(A, B) \longrightarrow \gcd(A-B, B)$ // say $A - B > B$

$\gcd(A, B) \longrightarrow \gcd(A-2B, B)$ // say $A - 2B > B$

$\gcd(A, B) \longrightarrow \gcd(A-3B, B)$ // say $A - 3B > B$

$\gcd(A, B) \longrightarrow \gcd(A-4B, B)$ // say $A - 4B > B$

⋮

$\gcd(\boxed{A - y^*B}, B)$ // Say $A - y^*B < B$

// y^*B is greatest multiple of B which we can subtract from A

$\gcd(A, B) \longrightarrow \gcd(A\%B, B)$

//Given $A >= B$ & $B > 0$

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

$$\gcd(\overset{a>b}{24, 16}) = \gcd(\overset{a<b}{8, 16}) = \gcd(\overset{a\%b}{8}, \overset{b}{16}) \longrightarrow \times$$

[$gcd(A, B) = gcd(B, A \% B)$] It will always work $a > b$ || $b < a$

$gcd(\overset{a}{24}, \overset{b}{16}) = gcd(\overset{a}{16}, \overset{b}{8}) = gcd(\overset{a}{8}, \boxed{\overset{b}{0}}) = ans = 8$

$gcd(\overset{a}{14}, \overset{b}{21}) = gcd(\overset{a}{21}, \overset{b}{14}) = gcd(\overset{a}{14}, \overset{b}{7}) = gcd(\overset{a}{7}, \boxed{\overset{b}{0}}) = ans = 7$

$a < b \xrightarrow[\text{flipped}]{\text{data}} a > b$

$gcd(\overset{a}{0}, \overset{b}{8}) = gcd(\overset{a}{8}, \boxed{\overset{b}{0}}) = ans = 8$

$gcd(\overset{a}{23}, \overset{b}{5}) = gcd(\overset{a}{5}, \overset{b}{3}) = gcd(\overset{a}{3}, \overset{b}{2}) = gcd(\overset{a}{2}, \overset{b}{1}) = gcd(\overset{a}{1}, \boxed{\overset{b}{0}})$

$= ans = 1$

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

TC: $O\left[\log_2^{\max(a,b)}\right]$

[Note: according to our function $gcd(0,0) = \boxed{0}$

We need to return according to Question]

```
main(){
    gcd(|a|, |b|) // Before applying gcd take absolute
}
```

Say $a < b \to$

$gcd(\overset{a}{14}, \overset{b}{21}) = gcd(\overset{a}{21}, \overset{b}{14}) = gcd(\overset{b}{14}, \overset{a\%b}{7}) = \dots$

$gcd(\overset{a}{-10}, \overset{b}{-15}) = gcd(\overset{|a|}{10}, \overset{|b|}{15}) =$

Say: 10:35

1Q) Given ar[N] calulate gcd of entire array

ar[3] = { 6 12 15 } :

ans = 6 6 3 : ans = 3

ar[4] = { 8 16 12 10 } :

ans = 8 8 4 2 : ans = 2

```
int  gcdarr ( int ar[], int n) {        TC: [n-1] * log_2 (max of array)
    ans = ar[0]
    i = 1; i < n; i++) {
        ans = gcd(ans, ar[i])
    }
    return ans;
}
```

$\boxed{TC \Rightarrow O\left[N \log_2 \text{max of array} \right]}$

Dry run : ar[4] =

0	1	2	3
8	16	12	10

ans = ar[0] = 8

i	ans = gcd (ans, ar[i])
1	ans = gcd (8, ar[1]) = gcd (8, 16) = 8
2	ans = gcd (8, ar[2]) = gcd (8, 12) = 4
3	ans = gcd (4, ar[3]) = gcd (4, 10) = 2
4	: return ans = 2

2Q) Given an array, check if there exists a subsequence with gcd = 1
↳ boolean

ar[5] = {4 6 8 8 3} : sub-seq : {4, 3} : return True

ar[5] = {16 10 6 15 27} : sub_seq : {16, 6, 15} : return True

ar[4] = {6 12 3 18} : no sub_seq : return false

Idea: Generate all subseq & Calulate gcd for each of them
↳ TC: $2^n * \{n \log_2^{max}\} \rightarrow 2^n * n \log_2^{max}$

idea2: Say 8 elements A B C D E F G H
↳ given subsequence B, D, F, G gcd(B, D, F, G) = 1
↳ gcd of entire arr[] = gcd(A B C D E F G H)
↳ gcd(gcd(B, D, F, G) gcd(A, C, E, H))
↳ gcd(1, gcd(A, C, E, H)) = 1

Claim: If there exist a sub-seq with gcd = 1
⇒ gcd of entire array = 1

↳ Final:
→ gcd of entire arr[] = 1 : Subsequ there, return True
→ gcd of entire arr[] ≠ 1 : No Subsequence with gcd = 1
return False

$TC: O(N \log_2^{max})$ → max of arr[]

Deleteone:

Given ar[N] elements, we have to delete 1 Element, such that gcd of remaining array elements is max

↳ find max gcd?

Ex:

gcd of rem elements

0*	1	2	3	4	
24	16	18	30	15	= 1

0	1*	2	3	4	
24	16	18	30	15	= 3

final ans = 3

0	1	2	3	4	
24	16	18	30	15	= 1

0	1	2	3	4	
24	16	18	30	15	= 1

0	1	2	3	4	
24	16	18	30	15	= 2

Idea: Repeat below appraoch for every element

delete an ar[i] element, & calulate gcd of all remaining elements, & get overall max

TC: $N * \{ \underbrace{[N-1] * \log_2^{max}}_{\text{TC to calulate gcd of N-1 elements}} \} \approx O(N^2 \log_2^{max})$

$\boxed{N>=2}$

```
int deleteOne(int ar[], int n){
    pfgcd[n] // pfgcd[i] = gcd of all elements [0 i]  TODO
    sfgcd[n] // sfgcd[i] = gcd of all elements [i, n-1]  TODO

    //Edge Cases:
    Delete ar[0]: gcd of all elements [1, n-1] = sfgcd[1]
    Delete ar[n-1]: gcd of all elements [0, n-2] = pfgcd[n-2]

    ans = max(sfgcd[1], pfgcd[n-2])

    i=1 ; i < n-1; i++){
        // delete ar[i] & get gcd of all remaining elements

        // ar[N]: [a0 a1 a2 ... a(i-1)] [a(i)] [a(i+1) a(i+2) ... a(N-1)]
                  <------- left ------>   *   <-------- right -------->

        left = gcd of all elements from [0, i-1]
        left = pfgcd[i-1] // gcd of all [0, i-1], i=0 pf[-1] err

        right = gcd of all elements from [i+1, N-1]
        right = sfgcd[i+1] // gcd of all [i+1, N-1], i=n-1, pf[n] err

        val = gcd(left, right) // gcd of all arr[] element
                                  except ar[i^th] element
        ans = max(ans, val)
    }
    return ans
}
```

TC: $\underbrace{N\log_2^{max}}_{pfgcd[\,]} + \underbrace{N\log_2^{max}}_{sfgcd[\,]} + \underbrace{N[\log_2^{max}]}_{\substack{i=1;\ i<n-1;\ i++)\{ \\ gcd(left, right)}}$

SC: $O(\underset{pfgcd[\,]}{N} + \underset{sfgcd[\,]}{N})$

Pseudo Code:

TC: $N\log_2 max$

```
int ar[N],
int pfgcd[N]
ans = 0;
i=0; i<n; i++){
    ans = gcd(ans, ar[i])
    pfgcd[i]=ans
}
```

```
int sfgcd[N] {Dry Run TODO}
ans = 0
i = n-1; i>=0; i--){
    ans = gcd(ans, ar[i])
    sfgcd[i]=ans
}
```

ar[4] = { 18 24 30 21 }

ans = 0

pfgcd[4] = { 18 6 6 3 }

i	ans = gcd(ans, ar[i])		pfgcd[i] = ans
0	ans = gcd(0, ar[0])	ans = ar[0] = 18	pfgcd[0] = ans
1	ans = gcd(18, ar[1])	ans = 6	pfgcd[1] = ans
2	ans = gcd(6, ar[2])	ans = 6	pfgcd[2] = ans
3	ans = gcd(6, ar[3])	ans = 3	pfgcd[3] = 3

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

$\boxed{TC: \log_2^{max(a,b)}}$

$\boxed{SC: \log_2^{max(a,b)}}$

$a >= b$: $gcd(a, b) = gcd(a\%b, b)$

1 $\underline{b <= a/2}$ $\frac{a}{2}$ $\underline{b > a/2}$ a

Case I : $b <= a/2$

$\underbrace{a\%b}_{[0,\ b-1]} < b <= a/2$

$a\%b <= a/2$

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

$a \rightarrow r = \boxed{\frac{a}{2}}$

$a \rightarrow a/2 \rightarrow a/4 \rightarrow a/8 \cdots$

$\log_2^a$ iterations

Case-II : $b > a/2$

$2b > a$

$2b - a > 0$

mul -1 in both sides

$a - 2b < 0$

add a in both sides

$2a - 2b < a$

$(a - b) < \frac{a}{2}$ ✓

$a - 2b < \frac{a}{2}$ ✓

$a - 3b < \frac{a}{2}$ ✓

$a - 4b < a/2$ ✓

$\vdots$

$a - \boxed{xb} < a/2$ ✓

max multiple of b less than <= a

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

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

$gcd(a, b) = gcd(< a/2, b)$

$a \rightarrow a/2 \rightarrow a/4 \rightarrow \cdots 1$

$\log_2^a$ iterations