Today's Content:

→ Gcd Intro

→ Properties of GCD

→ GCD function

→ gcd problems?

- Check Subsequence with gcd = 1
- Delete 1 element such that gcd of remaining elements in your array is max
- Given N distinct elements, pick 2 elements (i, j) & i != j, add their difference find min in array element
- If time permits more problems

GCD : Greatest Common divisor or **HCF** {Higher Common Factor}

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

↳ $n \rightarrow a \% n == 0$ && $b \% n == 0$, **n higher factor**

$gcd(15, 25) = 8$

↓	↓
1	1
3	**5**
5	25
15	

$gcd(12, 30) = 8$

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

$gcd(10, -25) = 8$

↓	↓
1	-25
2	-5
5	-1
10	1
	5
	25

→ -ve factors

$gcd(0, 8) = 8$

↓	↓
1	1
2	2
⋮	4
8	**8**
⋮	
∞	

$0 \% 1 == 0$

$0 \% 2 == 0$

$0 \% 3 == 0$

⋮

$0 \% n == 0$

$gcd(0, -10) = 10$

↓	↓
1	1
2	2
3	5
⋮	
10	**10**
⋮	
∞	

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

↓	↓
1	1
2	2
4	3
8	4
16	6
	8
	12
	24

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

↓	↓
1	**1**
2	3

$gcd(0, 0)$ = **not defined**

↓	↓
1	1
2	2
3	3
⋮	⋮
∞	**∞**

If given in Q :

They will tell you what to return

| | → abs

Properties of gcd(a, b)

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

→ $\gcd(a, b) = \gcd(|a|, |b|)$

$\gcd(0, n) = |n|$ $\gcd(0, 5) = 5$

↳ $n \neq 0$ $\gcd(0, 10) = 10$

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

↳ associative

Special property:

$A, B > 0$ & $A > B$

Say $\boxed{\gcd(A, B) = n}$ → $(A \% n == 0)$ & $(B \% n == 0)$

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

$(A - B) \% n = 0$ ✓ $B \% n = 0$ ✓

$(\underbrace{A \% n}_{0} - \underbrace{B \% n}_{0} + n) \% n$

$(0 - 0 + n) \% n = 0$

$B > A$:

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

Eg:

a b

gcd(23, 5) → gcd(18, 5) → gcd(13, 5) → gcd(8, 5) → gcd(3, 5)

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

given A, B ≥ 0   A > B

gcd(A, B) = gcd(B, A − B)

= gcd(B, A − 2B)

= gcd(B, A − 3B)

⋮

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

D₁ Q Divs

max no. of time we can subtract B from A

A > B   B < B

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

→ left side > right side

Note: If both 0, handle separately

```
//int gcd(int a, int b){
    if(b==0) {return a}
    if(a==0) {return b}
    // no need

    return gcd(b, a%b)
}
```

a b

gcd(15, 6)

↓ 15%6

return gcd(6, 3)

↓ 6%3

return gcd(3, 0)

↓

3

A B
gcd(6, 21)
6%21
A B
gcd(21, 6)
21%6
A B
gcd(6, 3)
A B
gcd(3, 0)

3
3
3

gcd(7, 0) : return 7

A B
gcd(0, 8)
A%B
A B
gcd(8, 0)

8
8

A > B

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

// Correct Code

```
int gcd(a, 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)
gcd(3, 6)
⋮

∞ loop
∞ recursion
Stack overflow

// gcd(|a|, |b|) → To avoid negative cases

// TC → $O(\log_2(\max(A,B)))$

Ex1: N → N-1 → N-2 → --- 1 : N iterations

Ex2: N → N/2 → N/4 → N/8 --- 1 : $\log_2 N$ iterations

$a \geq b$ }

not code // $gcd(a, b) = gcd(\underbrace{a \% b}_{\{< a/2\}}, b) \to O(\log_2 \max(A,B))$

// b, a

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

$a \% b < b < a/2$

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

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

$a \% b < [b = a/2]$

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

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

$\boxed{2b > a}$

$a \% b = \{ \boxed{a - b},\ a - 2b < 0 \ast \}$

$2b - a > 0$

//(-) on both sides

$a - 2b < 0$

// add a both sides

$2a - 2b < a$

$2(a-b) < a$

$\boxed{(a-b) < a/2}$

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

Case 3: How many times we can sub b

$a \% b = a - b$: (we can do only)

$a - 2b \ast$

{ Break : 10:40 break }

// given N array elements calculate gcd of Entire array

ar[3] → { 6 12 15 } → 15

↳ gcd of entire array :

→ gcd(max, min) ≠ gcd(arr)

Ex: ar[3] → { 5 6 10 }

gcd(5, 10) = 5

6 → 3

ar[4] → { 8 16 12 10 }

8

12 :

4 —— 10 : 2

gcd = 0

gcd = 1

gcd(0, 5) = 5

gcd(1, 5) = 1

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

gcd(5, 5) = 5

gcd(10, 10) = 10

ans = ar[0]

i=0

ans = gcd(ar[0], ans)

ans = ar[0]

```
int gcdarr(int ar[], int N){
    int ans = 0; / ans=ar[0]
    i=0; i<N; i++){
        ans = gcd(ans, ar[i])
    }
    return ans;
}
```

↳ Implemented

TC: N × ($\log_2$ max array elem)

TC: O(N)

gcd of Entire array: O(N)

TODO :

Hint1: for every gcd we use same ans variable

Hint2: ans ⇄ either same or decrease

I will discuss in next session

// Subsequence Basics

none or more

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

	0	1	2	3	4	5	6	7
ar[8] =	3	2	1	6	4	8	10	9

: {3 2 6 10 9} ✓

: {2 6 10 9 3} ✓ (order matters)

ar[4] = {6 9 0 8}

: {9 8} → ✓

: {6 9 8} → ✓

: {0 9 8} ✗ → order is matter

: {6 9 0 8} → ✓

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

bool : (T/F)

{Cannot Consider → { }}

Ex1: ar[5] = {4 6 3 8} → {4 3 8}

Ex2: ar[5] = {16 10 6 15 27} → {16 10 27}

Ex3: ar[4] = {6 12 3 18} → {No Subsequence}

Idea1: for every subsequence get gcd of Entire == 1

TC: (2^N * (TC to get gcd of Entire Subsequence))

Ex4: {3 6} → (No Subsequence)

Subsequence gcd = 1

Idea: ar[6] = A B C D E F }   gcd(B, C, F) = 1

// gcd of Entire array

= gcd(A, B, C, D, E, F) = gcd(gcd(A, D, E), $\underbrace{gcd(B, C, F)}_{1}$)

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

= $\underline{1}$

final obs:

→ If there is a Subsequence with gcd = 1

↳ gcd of Entire array == 1 : (Subsequence ✓)

≠ 1 : (Subsequence ✗)

Delete one:

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

0	1	2	3	4
24	16	18	30	15
*				

gcd( ) : 1

0	1	2	3	4
24	16	18	30	15
	*			

: 3 → ans

gcd( )

0	1	2	3	4
24	16	18	30	15
		*		

: 1

gcd( )

0	1	2	3	4
24	16	18	30	15
			*	

: 1

0	1	2	3	4
24	16	18	30	15
				*

: 2

gcd( )

// gcd of Entire array:

// Skip i^{th} Elem = $\frac{gcd(\ )}{ar[i]}$ *

Idea1: Remove every element & get gcd of Entire array

TC: (N * [get gcd of Entire array] → $O(N^2)$
$\underbrace{\text{get gcd of Entire array}}_{O(N)}$

// While taking gcd, skip the element we want to remove

// Assume N = 7: 0 1 2 3 4 5 6

Delete	gcd:
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])

// Pfgcd[i] = gcd of all Elements [0 i]

// Sfgcd[i] = gcd of all Elements [i N-1]

Optimization:

// Construct Pfgcd[N];

Pfgcd[0] = ar[0]

i=1; i< N; i=i+1){

Pfgcd[i] = gcd (Pfgcd[i-1], ar[i])

[0 i] ⟷ [0, i-1] ⟷ [i]

}

// Construct Sfgcd[N]

Sfgcd[N-1] = ar[N-1]

i = N-2; i>=0; i--){

Sfgcd[i] = gcd (Sfgcd[i+1], ar[i])

gcd[i, N-1] ⟵ gcd[i+1, N-1] ⟺ [i]

}

// Delete Every element

ans = 0;

i = 0; i < N; i++) {

// delete i^{th}

left → [0, i-1] → Pfgcd[i-1] → if i==0 Edge Case

right → [i+1, N-1] → sfgcd[i+1] → if i==N-1 Edge Case

TODO: handle Edge Cases

ans = max(ans, gcd(left, right))

}

Rember Subtract : % → gcd

$A \% M = B \% M$

$A \% M - B \% M = 0$

$A \% M - B \% M + M = M$

Apply % M on both side

$(A \% M - B \% M + M) \% M = 0$

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

$M = A - B \quad : A > B$

$B - A \quad : B > A$

3Q) Pubg: {TODO} → Hint: (Repeated Subtraction)

N players in pubg. Each player has initial health A[i]

If player i attacks j → {Health not effected to person who attacks}

```
if( A[i] >= A[j]){
    player j dies
    A[j] = 0
}

if( A[i] < A[j]) {
    A[j] = A[j] - A[i]
}
```

Ex:

A	B
6	10

Q) find min health of last person standing

{6, 4}

Observation :

Conclusion :

Ex: {12, 26}

$\downarrow$

Ex:

{6 4 8 3