Maths GCD

Content

- GCD Intro
- Properties of GCD
- GCD optimised code

- Delete One Element s.t. GCD of remaining element is max
- PUBG

GCD — Greatest Common Divisor

$$gcd(a, b) = x : \{ x \text{ is greatest no. s.t. } a \% x == b \% x == 0 \}$$

gcd (15, 25)

1	1
3	5
5	25
15	

gcd (12, 30)

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

gcd (10, -25)

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

NOTE :
Since we are int.... gcd we only care about positive factors

gcd (0, 8)

1	1
2	2
⋮	4
⋮	8
∞	

gcd (0, -10)

1	1
2	2
⋮	5
	10
∞	

gcd (-16, -24)

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

gcd (-2, -3)

1	1
2	3

gcd (0, 0)

cannot be defined

Bruteforce Pseudocode

TC: O(max)

```
int bruteGCD(a, b) {
    int max = min(|a|, |b|)

    for (x → max to 1) {
        if (a % x == 0 && b % x == 0) {
            return x
        }
    }

    return 0
}
```

Properties of GCD

- $\rightarrow \gcd(a, b) = \gcd(b, a)$ {commutative}
- $\rightarrow \gcd(a, b) = \gcd(|a|, |b|)$
- $\rightarrow \gcd(0, x) = |x|$ if $x != 0$
- $\rightarrow \gcd(a, b, c) = \gcd(\gcd(a, b), c) = \gcd(\gcd(b, c), a) = \gcd(\gcd(a, c), b)$ } Associative

Special Property

Given

$A, B > 0$ and $A >= B$ and $\gcd(A, B) = x$

Given $\rightarrow$ $\gcd(A, B) = x$ $\quad A \% x == 0$, $B \% x == 0$

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

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

Expand %

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

$A \% x \rightarrow 0$, $B \% x \rightarrow 0$, $(x) \% x \rightarrow 0$

$\Rightarrow$ LHS == RHS

$\gcd(A, B) = \gcd(A - B, B)$ $\quad$ $A >= B$

Eg: gcd (23, 5) (a = 23, b = 5)

$= gcd(23-5, 5)$

$= gcd(23-10, 5)$

$= gcd(23-15, 5)$

$= gcd(23-20, 5)$

$= gcd(3, 5)$

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

$A >= B$

$gcd(A-2B, B)$

$\vdots$

$gcd(A-iB, B)$

largest multiple of B such that $iB < A$

$gcd(A \% B, B)$

$gcd(A, B) = gcd(A \% B, B)$ Case i

$gcd(A, B) = gcd(B, A \% B)$ Case ii

```
int gcd (a, b) {          // case i
    // Base condition
    return gcd(a%b, b)
}
```

① → gcd(24, 16)
② → gcd(24%16, 16)
→ gcd(8, 16)
③ gcd(8%16, 16)
gcd(8, 16)
...

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

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

gcd(24, 16)
→ gcd(16, 24%16)
→ gcd(16, 8)
→ gcd(8, 16%8)
→ gcd(8, 0)

gcd(24, 16) → 8
gcd(21, 14%21) → gcd(21, 14)
gcd(14, 21%14) → gcd(14, 7)
→ gcd(7, 0)
gcd(0, 8) → gcd(8, 0%8) = gcd(8, 0)

gcd(23, 5) → gcd(5, 3) → gcd(3, 2)
→ gcd(2, 1) → gcd(1, 0)

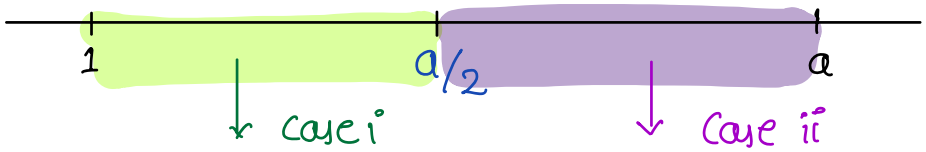
NOTE: gcd(0,0) → 0

Never asked

Time Complexity GCD

gcd (a, b) $a >= b$

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



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

$\Rightarrow a \% b < b$

$a \% b$: $(0, b-1)$

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

$a \longrightarrow \frac{a}{2} \longrightarrow \frac{a}{4} \ldots\ldots 1$

k

$k = \log a$

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

$2b > a$

$0 > a - 2b$

$a > a + a - 2b$

$a > 2a - 2b$

$a > 2(a-b)$

$a/2 > a - b$

$a - b < a/2$

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

$a \longrightarrow \frac{a}{2} \longrightarrow \frac{a}{4} \ldots\ldots \ldots 1$

$\log(a)$

TC: $gcd(a, b) = O(\log_2 \max(a, b))$

Q> Given A[N] calculate gcd of entire array.

A[3] = { 6 12 15 }

(6 12)

(6 , 15) → (3)

A[4] = { 8 16 12 10 }

8 2

(2)

```
int gcdAll ( A[] ) {
    ans = A[0]
    for ( i → 1 to n-1 ) {
        ans = gcd(ans, A[i])
    }
    return ans
}
```

TC: $O(N \log_2 max)$ max = max(A)

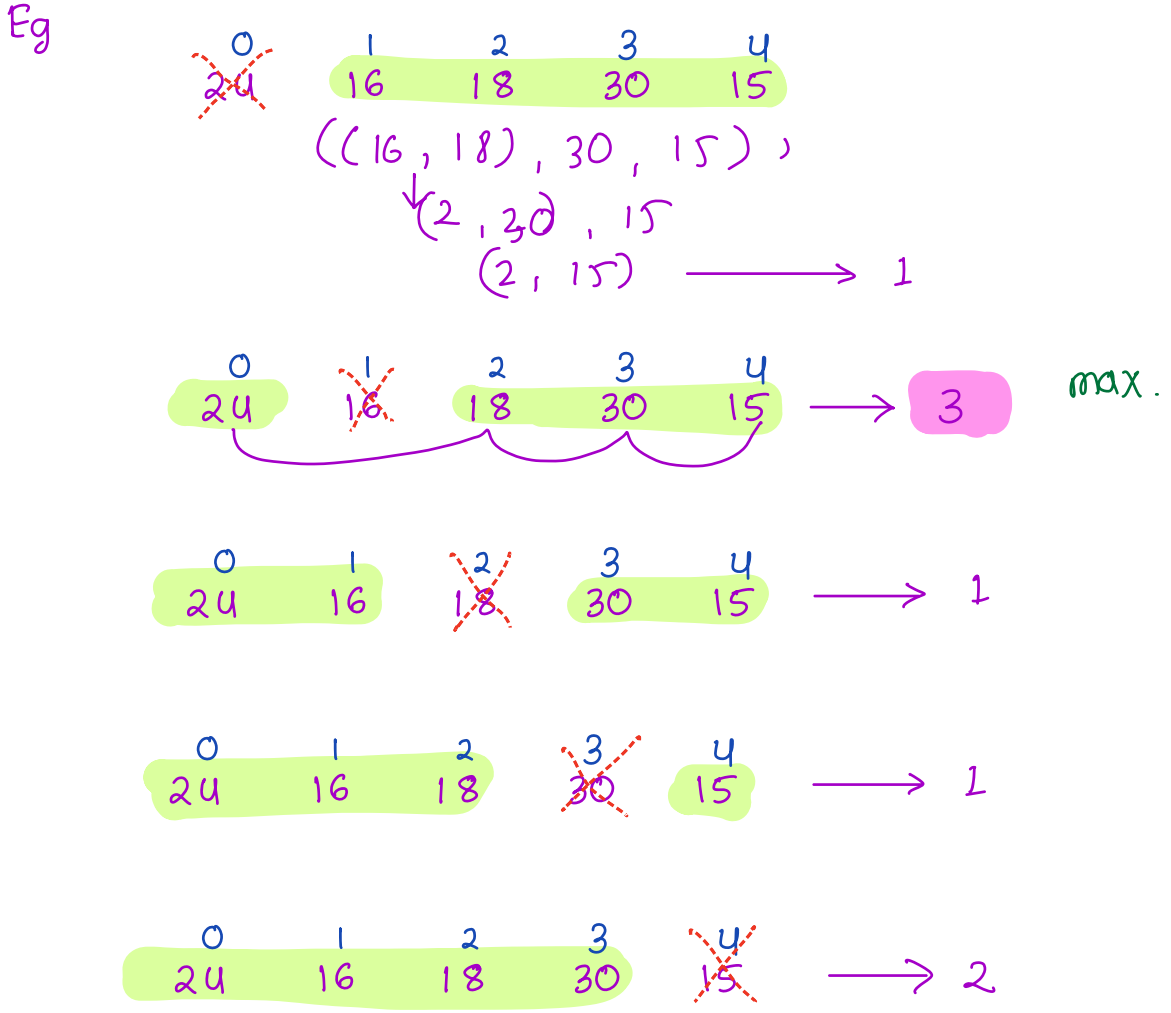
Break : 8:40 am.

Delete One

Given A[N] elements, we have to delete one element, such that gcd of remaining array is max

Find max gcd ? A[i] > 0

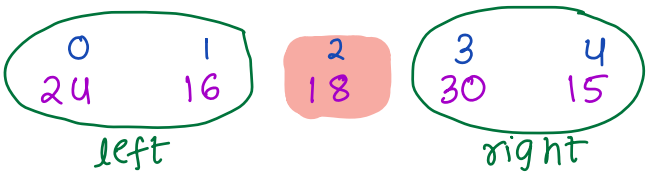
Eg



Bruteforce :

```
for i ⟶ 0 to n-1 {
    skip ith element and take GCD for rest
    max of all gcds
}
```

```
int deleteOne (A[]) {

    maxgcd = 0

    pfgcd [N]   //  ith - gcd of 0 to i
    sfgcd [N]   //  ith - gcd of i to n-1

    for (i → 0 to n-1) {
        ~~left = gcdAll (0, i-1)~~
        left = 0 , right = 0  ⟶  note gcd(0,x) = |x|
        if (i > 0)
            left = pfgcd [i-1]

        ~~right = gcdAll (i+1, n-1)~~
        if (i < n-1)
            right = sfgcd (i+1)

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

    return maxgcd .
```



TC: O(N log max)

TC : O(N log max) + O(N log max) + O(N log max)

pfgcd sfgcd for loop

PUBG

Let N players are playing PUBG and

A[i] = health of i^{th} player.

If i^{th} player attacks j^{th} player

(i) ⟶ if $A[i] \leq A[j] \Rightarrow A[j] = A[j] - A[i]$

(ii) ⟶ else $A[j] = 0$ (die)

Find **min** health of the last surviving person.

$A = \{6, 4\} \longrightarrow$ 6 attacks 4 $\{6, 0\}$

$\longrightarrow$ 4 attacks 6 $\{6, 4\}$

$\longrightarrow \{2, 4\}$

$\longrightarrow \{2, 2\}$

$\longrightarrow \{2, 0\} \longrightarrow 2$

6

any = 2

$A = \{6, 10, 15\}$

$\Longrightarrow$ 6 attacks all ⟶ (6, 4, 9)

4 attacks all ⟶ (2, 4, 5)

2 attacks all ⟶ (2, 2, 3)

2 attacks all ⟶ (2, 0, 1)

1 attacks all ⟶ (1, 0, 1)

⟶ (0, 0, 1) ⟶ 1

any = 1

Note : we can minimize if player with min health attacks

[23, 5] ⟶ [23 − 5, 5]

⟶ [23 − 5 − 5, 5]

⟶ [23 − 5 − 5 − 5, 5]

⟶ [23 − 5 − 5 − 5 − 5, 5]

→ [3, 5]
→ [3, 5-3]
→ [3, 2]
→ [1, 2]
→ [1, 1]
→ [0, 1] → (1)

→ [23, 5, 10]

(5, 10) will fight
5 will survive

[23, 5] → (1)

Doubt session

GCD of PF

→ pfgcd [N]
ans = 0

```
for( i → 0 to n-1 ) {
    ans = gcd (ans, A[i])
    pfgcd[i] = ans
}
```

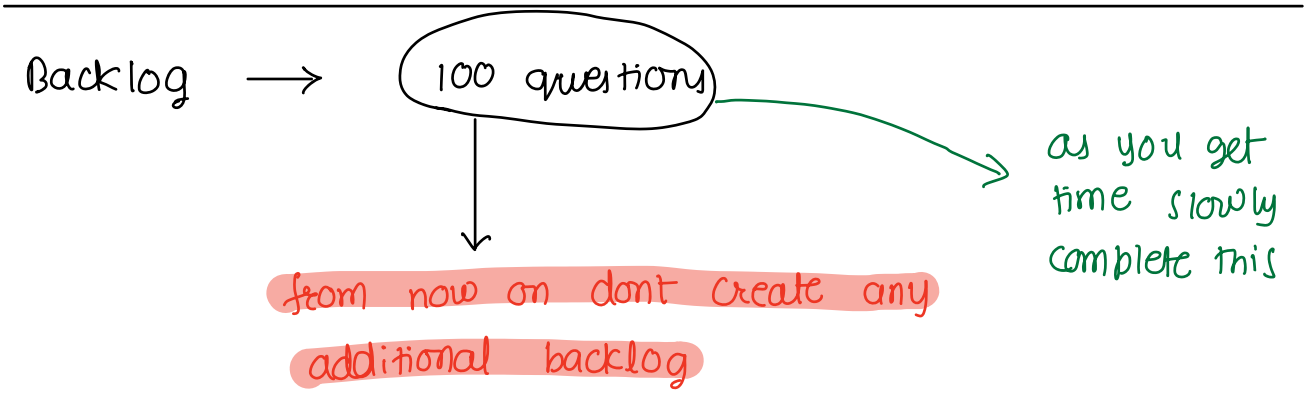
↓ ↓
2 3 5

ans = 0
ans = gcd(0,2)

2 1 1

(2,3)

```
sfgcd [N]
ans = 0

for( i → n-1 to 0 ) {
        ans = gcd (ans, A[i])
        sfgcd[i] = ans
}
```



25 mins to a single problem....