Maths - GCD

6-8'

GCD : Greatest Divisor

HCF : Highest Common Factor

Every factor $\le a$, $\le b$

$\le \min(a,b)$

$gcd(a,b) \Rightarrow$ the greatest number which divides both a and b

$15/5 = 3$
$25/5 = 5$

$gcd(15,25) = 5$

$gcd(7,9) = 1$

$gcd(7,14) = 7$

if $gcd(a,b) \Rightarrow$ one number should be prime

g

Q1) if $gcd(a,b) = 1 \Rightarrow$ one number has to be prime

$\boxed{gcd(20,21) = 1}$

x x

Q2) If one number is prime $\Rightarrow$ $gcd(a,b) = 1$

$gcd(7,14) = 7$

$gcd(5,15) = 5$

Max value of $gcd(a,b)$

Brute Force :

ans = 1;

for(i = 2; i ≤ min(a,b); i++) {

... b%i == 0)

for ...
if(a%i == 0 && ...
ans = i;

}

return i;

T.C: O(min(a,b))

Approach 2:

```
int ans;
for( i = min(a,b); i ≥ 1 ; i--){
    if( a % i == 0  &&  b % i == 0){
        ans = i;
        break;
    }
}
return ans;
```

T.C: O(min(a,b))

min(a, b) — 1

— 2

1

if gcd(a,b) = 1

(Co-Prime)

gcd (15, 21) = 3

gcd (9, 14) = 1

(P₁, P₂) = 1

LCM: Least Common Multiple

any Multiple ≥ a
≥ b

LCM(100, 75)

100 : 100, 200, 300, 400, 500 ...

75 : 75, 150, 225, 300, 375, ..

gcd (100, 75) = 25

LCM(50,75)

≥ a ≥ 50 — ①

≥ b ≥ 75 — ②

≥ max(a,b)

Min value of LCM(a, b) = ?

max (a, b)

min value =

Max Value = $a \times b$ → Max value of

Multiple of a   Multiple of b

ab   2ab

```
for(i = max(a,b); i <= a*b; i++){
    if(i%a == 0 && i%b == 0){
        return i;
    }
}
```

Relation between a, b, lcm(a,b), gcd(a,b)

$$a \times b = gcd(a,b) \times lcm(a,b)$$

$LCM(100, 75) = 300$

100 divides 300 ⇒ $100 \mid 300$

75 divides 300 ⇒ $75 \mid 300$

$x \mid y$ ⇒ x divides y

$gcd(100, 75) = 25$

$25 \mid 100$, $25 \mid 75$

Properties of GCD

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

2) $gcd(a, 0) =$

0 can be divided by any number

$0 \% i == 0 \quad \forall i > 0 \Rightarrow [1, \infty]$

Highest num which divides a $\Rightarrow$ a

$a \geq b$

$$\boxed{gcd(a,0) = a}$$

3) $gcd(a, 1) = 1$

Replaces the greatest number with $b-a$

4) $gcd(a, b) = gcd(a, b-a)$ when $b \geq a$

$gcd(b-a, a)$

$a = 7, b = 11$

$d = 1$

Let $\boxed{gcd(a,b) = d} \Rightarrow d | a$ & $d | b$

$a = k_1 d \qquad b = k_2 d$

$7 = k_1 \Rightarrow k_1$

$(11-7) = (k_2 - k_1)$

4

$b - a = (k_2 - k_1) d \qquad \{ b \geq a \Rightarrow k_2 \geq k_1$

$b - a = 0d$

$b \geq a$

$d | b-a$ — (1)

$k_1 = k_2$

$\Rightarrow d$ is a factor of $(b-a)$

if $gcd(b-a, a) = d$

Let $\boxed{gcd(a, b-a) = m} \Rightarrow m | a$ & $m | b-a$

$\Rightarrow a = t_1 m \qquad b-a = t_2 m$

$a + b - a = m(t_1 + t_2)$

$\Rightarrow b = m(t_1 + t_2) \Rightarrow m | b$ — (2)

and b

m divides

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

$$\boxed{m \le d}$$

m is a factor of a, b

d is the highest factor of a, b

$m \le d$ — 3

$\boxed{gcd(a,b) = d}$

$\Rightarrow$ $d \mid b-a$ & $d \mid a$

d is a factor of $(b-a, a)$

Greatest factor of $(b-a, a) = m$

$\boxed{d \le m}$ — 4

$\boxed{m \le d}$ — 3 and $\boxed{d \le m}$ — 4

$$\boxed{m == d}$$

$gcd(b-a, a)$ $gcd(a,b)$

$$\boxed{gcd(a,b) = gcd(b-a, a)}$$

---

5) $gcd(a,b) = gcd(a, b-a) = gcd(a, b-a-a)$ $b \ge a$

$= gcd(b-a-a-a)$

b%a

$\Rightarrow gcd(a, b-a-a \ldots a)$

till it is < a

$$\boxed{gcd(a,b) = gcd(a, b\%a)}$$

step 1

gcd (3, 20) = gcd (3, 20%3)

gcd (3, 2) → gcd (3%2, 2)

gcd (1, 2) step2

gcd (1, 2%1) step3

gcd (a, 0) = a

$\boxed{gcd(1, 0)}$ → $\boxed{1}$

Euclidean Algo for finding GCD

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

    if( a ≥ b) return gcd(a%b, b );
    else return gcd ( b%a, a );

}
```

b ≤ a

2 - line code ?

Assume always $\boxed{b \le a}$

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

gcd(6, 30) when a > b

$gcd(6, 30) \Rightarrow \boxed{gcd(30, 6)}$

a b $\quad b < a$

$6 \% 30$

Euclidean Algo for find GCD

$\_gcd(\ );$

$\boxed{gcd(30, 6)} \Rightarrow gcd(6, 30\%6)$

$\Rightarrow gcd(6, 0) \Rightarrow 6$

b

Time Complexity

if $a > b$, $a \% b \Rightarrow$ less than b

$a\%b < b$

$a \% b$

Case 1: $\frac{a}{2} < b$

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

$a = 2b$

$gcd(2b, b)$

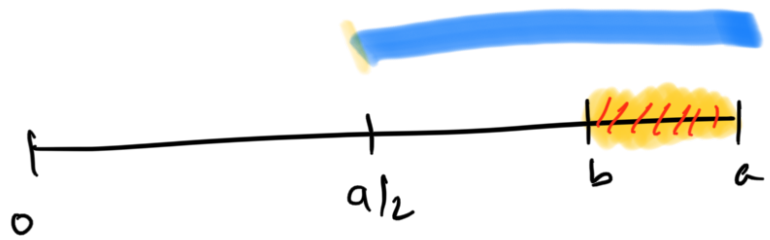
$gcd(a,b) = b = \frac{a}{2}$

$\boxed{a\%b = 0}$

Case 3: $b < \frac{a}{2}$

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

Case 1: $\frac{a}{2} < b$



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

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

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

subtraction of b from a)

(a%b is repeated sub...

→ In all 3 cases, the upper bound of a%b is $\frac{a}{2}$. if ($a \geq b$)

$$\underbrace{a \rightarrow \frac{a}{2} \rightarrow \frac{a}{2^2} \rightarrow \frac{a}{2^3} \rightarrow \frac{a}{2^4} \cdots\cdots}_{K \text{ steps}} \quad 1$$

$\frac{a}{2^1}$ $\frac{a}{2^2}$ $\frac{a}{2^3}$ . . . . $\frac{a}{2^K}$

$$\frac{a}{2^K} = 1$$

$$2^K = a$$

$$\boxed{K = \log(a)} \quad (a \geq b)$$

$$K = \log(\max(a,b))$$

$$P = \max(a,b)$$

$$\boxed{T.C: O(\log P)}$$

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

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

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

Question:

Given an integer array check if there is a subsequence with gcd = 1

... {}

A = [3, 7, 9, 1, 6]

(9, 1, 7) ✗

(3, 9, 1) ✓

(3, 9, 6) ✓

$\underbrace{2 \times 2 \times 2 \cdots 2}$ $\boxed{2^n}$

Non-Empty → $\boxed{2^n - 1}$

Total subsequence : $\boxed{2^n}$

→ A = (6, 10, 15)

(7, 12, 10, 15)

Brute Force

Consider all subsequences 2^n

$N = 10^5$

2^{10^5}

$\boxed{\text{Recursion / Back Tracking}}$ ⇒

{ 1 }

Better Approach :

Observations

1) If any number A[i] = 1, TRUE

2) If I have a pair of numbers, such that they are coprime ⇒ gcd(A[i], A[j]) = 1

A = $\boxed{[6, 10, 15]}$ ⇒ 1

gcd(6, 10) = 2

$gcd(10, 15) = 5$
$gcd(6, 15) = 3$

3) $gcd(a, 1) = 1$

6 elements

$A = [a_1, a_2, a_3, a_4, a_5, a_6]$

there is a subsequence s' which has $gcd = 1$

$[a_2, a_4, a_6]$

Let $\boxed{gcd(a_2, a_4, a_6) = 1}$

$gcd(a_2, a_3, a_4, a_6) = 1$

$gcd(a_3, gcd(a_2, a_4, a_6))$

$\Rightarrow gcd(a_3, 1) = \boxed{1}$

$gcd(a_1, a_2, a_3, a_4, a_5, a_6) = \boxed{1}$

$gcd(gcd(a_1, a_3, a_5), gcd(a_2, a_4, a_6))$

$\Rightarrow gcd(gcd(a_1, a_3, a_5), 1) = \boxed{1}$

$gcd(A) = ?$ $\boxed{1}$?

→ take GCD of whole Array
→ if gcd = 1, Return True
else return False

$gcd(a, b) \Rightarrow$
$\log(\max(a, b))$

A = 3 4 5 6 7 8

T.C: (N-1) times computing gcd

A = 7 6 7 6 7 6 7 6

$\boxed{\text{T.C: } N \times \log(\max(A))}$ → None of these

```
ans = A[0];
for (i = 1; i < n; i++) {
    ans = gcd(ans, A[i])
}
if (ans == 1) return True
else return False;
```

Question:

Given array of elements, delete exactly one element from the array such that gcd(array) is maximised.

→ A = 9 18 49 12 30

gcd(A) = 1

Remove 49

gcd(9, 18, 12, 30) = 3

→ A = 9 18 17 12 30

Remove $\boxed{17}$

gcd(9, 18, 17, 12)

[7 3 6 9 12] ✓

[7 15 45 30 60] ✓

[14 15 45 30 60] ✓

⇒ gc..

7 3 6 9 12

7 1

(7, 6, 9, 12) = 1

(3, 6, 9, 12) = $\boxed{3}$

Brute Force:

Consider deleting every element

$N \times (N \times \log(\max A))$

$T.C = O(N^2 \log(\max(A)))$ ⇒ Brute Force

$A = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7]$

Delete a_3 = $gcd(a_1, a_2, a_4, a_5, a_6, a_7)$

⇒ gcd (gcd(a_1, a_2), gcd(a_4, a_5, a_6, a_7))

gcd(a_1, a_2, a_3), gcd(a_5, a_6, a_7)

For any element i,

gcd(0..... i-1), gcd(i+1..... n-1)

Prefix GCD

Suffix GCD

A i, to be deleted

For any element :

gcd of Remaining element = $\boxed{gcd(pref[i-1], suff[i+1])}$

T.C : $O(N \times \log(max)) + N \times (\log(max)) + O(N)$

// consider deleting a[i]

```
for (i=0; i<n; i++) {
    ans = max(ans,
              gcd(pref[i-1], suf[i+1]))
}
```

$$\boxed{\begin{aligned} T.C &: O(N \times \log(max)) \\ S.C &: O(N) \end{aligned}}$$

Question : PUBG

there are N players with health A[i]

A = [3, 4, 1, 7]

If player i attacks player j (Health of j will decrease)

→ if(A[i] >= A[j]) => A[j] = 0 (j dies)

if(A[i] < A[j]) => A[j] = A[j] - A[i]

Find the minimum health of the last / remaining person.

A = [6, 4]

attacks 4 => [6, 0]

Case1: 6 ...

Case2: 4 attacks 6

[2, 4]

2 attacks 4 → [2, 2] → [$\boxed{2}$, 0]

4 attacks 2 → [0, $\boxed{4}$]

10 player

[10, 4] → [$\boxed{10}$, 0]

[6, 4] → [2, 4] → [2, 2] → [$\boxed{2}$, 0]

Observations

1) It's always better for a person of lower health to attack person of higher health

Because we want to minimise the health of last person

We have 2 players a, b   b > a

a → b

$(a, b) \to (a, b\%a)$

$b = \underbrace{b - a - a - a - a}_{\text{till it is less than } a} \Rightarrow \boxed{b\%a} \to \boxed{x}$ Health of b

b → a

$a = \underbrace{a - x - x - x \dots}_{\text{till it's} < a} \Rightarrow a\%x$

Min Health of last Person = $\boxed{gcd(a, b)}$

$A = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5]$

$\boxed{\text{min Health} = \gcd(A)}$

T.C: $N \times \log(\max(A))$

S.C: $O(1)$

$A = [6, 4, 8]$

$[2, 4, 8]$

$[2, 2, 8]$

$[0, 2, 8]$

$\boxed{[0, 2, 0]}$

$\boxed{2}$

8 % 2

8 − 2 − 2 − 2 .

< 2

8 % 2

P1 P2 P3

$\boxed{d = \gcd(P_1, P_2)}$

$\text{Min Health} = \gcd(d, P_3)$

$= \boxed{\gcd(\gcd(P_1, P_2), P_3)}$

$x = b \% a$

$(b \% a, a) \rightarrow (x, a \% x)$

(a, b)

(n, a)

- Euclidean Algo

n < a

T.C: $\boxed{\log_2(N)}$

$N = 2^{32}$

$32(\log_2 2) = \boxed{32}$

Only 32 operations

(10^{18})

$N = 2^4$

steps = 4

$O(1)$

32 operations