

Number Theory Lecture 3

Sunday, 7 July 2024 5:32 PM

Greatest Common Divisor (GCD) = HCF
 ↳ Highest Common Factor.

$$\gcd(12, 18) = ? \text{ (6)}$$

$$\begin{array}{r|l} 2 & 12 \\ \hline 2 & 6 \\ \hline 3 & 3 \\ \hline & 1 \end{array}$$

$$12 = 2^2 \times 3^1 \quad \checkmark$$

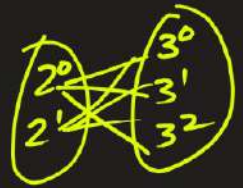
$$18 = 2^1 \times 3^2 \quad \checkmark$$

$$\gcd(12, 18) = 2^1 \times 3^1$$

$$\begin{array}{r|l} 2 & 18 \\ \hline 3 & 9 \\ \hline 3 & 3 \\ \hline & 1 \end{array}$$

$$\text{or } 2^0 \times 3^0 \text{ or } 2^1 \times 3^0 \text{ or } 2^0 \times 3^1 \text{ or } 2^1 \times 3^1$$

$$\begin{aligned} \# \text{divisors of } 18 \\ &= 2 \times 3 \\ &= 6 \end{aligned}$$



Divisors of 12:-

Divisors of 18:-

$$2^0 \times 3^0 = 1$$

$$2^0 \times 3^0 = 1$$

$$2^1 \times 3^0 = 2$$

$$2^0 \times 3^1 = 3$$

$$2^2 \times 3^0 = 4$$

$$2^0 \times 3^2 = 9$$

$$2^0 \times 3^1 = 3$$

$$2^1 \times 3^0 = 2$$

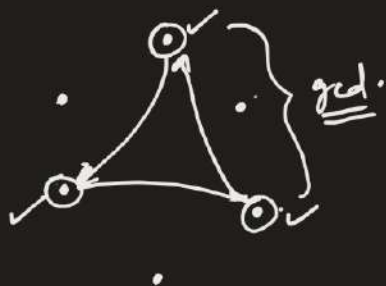
$$2^1 \times 3^1 = 6$$

$$2^1 \times 3^1 = 6$$

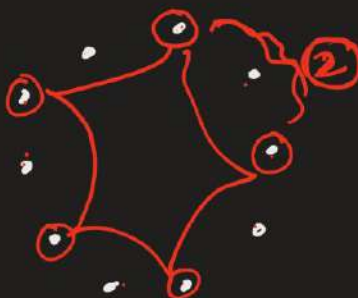
$$2^2 \times 3^1 = 12$$

$$2^1 \times 3^2 = 18$$

$$\gcd(\underline{4}, \underline{6})$$



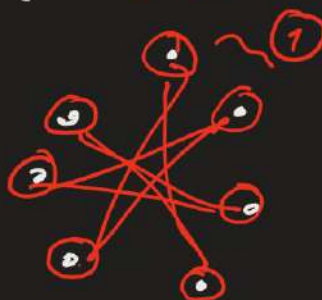
$$\gcd(\underline{8}, \underline{10})$$



$$\gcd(4, 8)$$



$$\gcd(\underline{7}, \underline{3})$$



$$\bullet \gcd(a, 1) = 1$$

$$\underline{\underline{a > 0}}$$

—x—

Q. How to find gcd of a and b?

```
    a > b
for (i: b → 1) {
    if (a % i == 0 && b % i == 0) {
        gcd = i;
        break;
    }
}
```

$O(b)$
 \equiv
 $O(\min(a, b))$

• $\gcd(a, b) = \gcd(\underline{a-b}, b)$, $\underline{a \geq b}$

Proof: $\gcd(a, b) = g$

$\Rightarrow \underline{a} \mid g$ and $\underline{b} \mid g$

$\Rightarrow \underline{a-b} \mid g = \underbrace{\left(\frac{a}{g}\right) - \left(\frac{b}{g}\right)}$

• $\gcd(a, 1) = 1$ $a > 0$

• $\gcd(a, 0) = a$

• $\gcd(\underline{a}, b) \leq \min(a, b)$

• $\gcd(a, a) = \underline{a}$

• $\gcd(a, b) = \gcd(b, a)$

Euclidean Algorithm to find gcd

- $\gcd(a, b) = \gcd(b, a-b)$
- $\gcd(a, 0) = a$

$\gcd(\underline{a}, \underline{b})$ {
if $b == 0$ return a ;

$O(b)$

{ if $(\underline{a-b} > b)$ return $\gcd(a-b, b)$;
else return $\gcd(b, a-b)$;

}

Time = $O(a)$

$$\begin{aligned}\gcd(18, 12) \\&= \gcd(12, \underline{6}) \\&= \gcd(6, 6) \\&= \gcd(6, 0) \\&= \boxed{6}\end{aligned}$$

$$\begin{aligned}\gcd(8, \underline{1}) \\&= \gcd(7, 1) \\&= \gcd(6, 1) \\&= \gcd(5, 1)\end{aligned}$$

$$\begin{aligned}&\vdots \\&= \gcd(1, 1) \\&= \gcd(1, 0) \\&= 0\end{aligned}$$

$\underline{a-b} < 0$
if $\underline{b > a}$

$$\begin{aligned}\gcd(\underline{12}, \underline{10}) \\&= \gcd(\underline{10}, \underline{2}) \\&= \gcd(8, 2) \\&= \gcd(6, 2) \\&= \gcd(4, 2) \\&= \gcd(2, 2) \\&= \gcd(2, 0) \\&= \underline{(2)}\end{aligned}$$

$$\begin{aligned}\gcd(19, 11) \\&= \gcd(11, 8) \\&= \gcd(8, 3) \\&= \gcd(5, 3) \\&= \gcd(3, 2) \\&= \gcd(2, 1) \\&= \gcd(1, 1) \\&= \gcd(1, 0) \\&= \underline{(1)}\end{aligned}$$

Optimized Euclidean algorithm

• $\gcd(\underline{a}, \underline{b}) = \gcd(a-b, b) \quad a > b$

$$\gcd(\underline{a-b}, b) \quad a-b > b$$

$$= \gcd(\underline{a-2b}, b) \quad a-2b > b$$

$$= \gcd(a-3b, b) \quad a-3b > b$$

⋮

$$= \gcd(\underline{a - \lfloor \frac{a}{b} \rfloor \cdot b}, b)$$

$$\gcd(12, 8)$$

$$= \gcd(8, 4)$$

$$\lfloor \frac{12}{8} \rfloor = 1$$

12

$$\gcd(12, 2)$$

6

How many times can I subtract b from a
 $\lfloor \frac{a}{b} \rfloor$ times

$$\gcd(12, 5)$$

$$= \gcd(7, 5)$$

$$= \gcd(5, 2)$$

$$\lfloor \frac{12}{5} \rfloor = 2$$

$$\lfloor \frac{a}{b} \rfloor \cdot b$$

$$\gcd(16, 3)$$

$$= \gcd(16-3, 3)$$

$$= \gcd(16-3 \times 2, 3)$$

$$= \gcd(16-3 \times 3, 3)$$

$$= \gcd(16-3 \times 4, 3)$$

$$= \gcd(\underline{16-3 \times 5}, 3)$$

1

$$\lfloor \frac{16}{3} \rfloor = 5$$

$$\underline{\text{gcd}(a, b)} = \text{gcd}(\underline{a - \lfloor \frac{a}{b} \rfloor \cdot b}, b) \\ = \text{gcd}(a \% b, b)$$

$$\begin{aligned} \text{gcd}(16, 3) \\ = \text{gcd}(16 - \lfloor \frac{16}{3} \rfloor \cdot 3, 3) \\ = \text{gcd}(16 - 15, 3) \\ = \text{gcd}(1, 3) \\ \underline{16 \% 3} = \textcircled{1} \end{aligned}$$

$$\underline{a \% b < b}$$

$$\begin{cases} \text{gcd}(\underline{a}, \underline{b}) = \text{gcd}(\underline{b}, \underline{a \% b}) = \text{gcd}(a \% b, b \% (a \% b)) \dots \\ \text{gcd}(a, 0) = a \end{cases}$$

$$\begin{cases} \text{gcd}(a, b) \{ \\ \text{if } (b == 0) \text{ return } a; \\ \text{return } (b, a \% b); \} \end{cases}$$

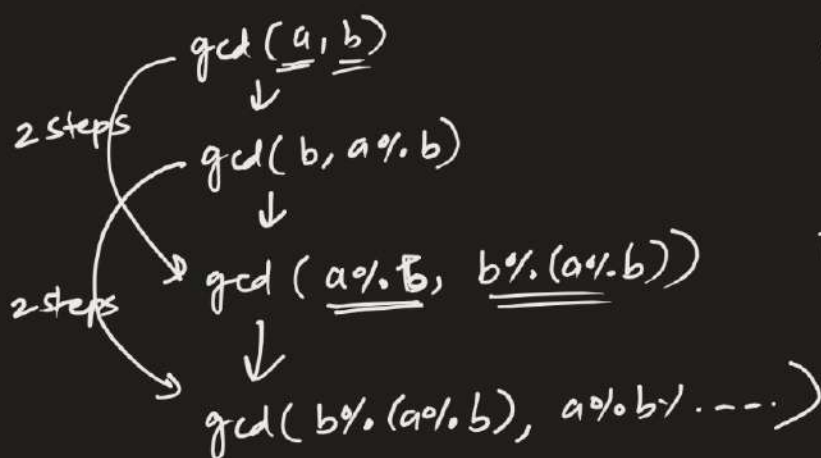
$$\text{Time} = O(\quad)$$

$$a \% b \leq a/2$$

$$\min(a, b) = \textcircled{b}$$

$$T(n) = T(n/2) + O(1)$$

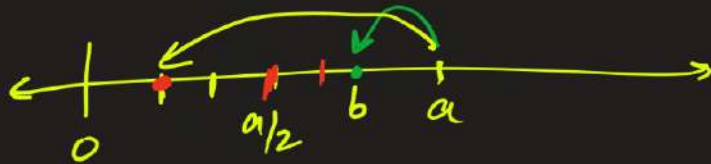
$$\underline{T(n) = O(\log n)} \\ O(\log b)$$



$$\text{gcd}(8, 5)$$

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

$$a \% b \leq a/2$$



$$\begin{aligned} \gcd(8, 5) &= \gcd(5, 3) \\ &= \gcd(3, 2) \\ &= \end{aligned}$$

x

Linear Diophantine Equations

Find integral solution to the equation

$$ax + by = c$$

/ / /
Integers

$$\begin{cases} x = - \\ y = - \end{cases} \quad x, y \in \mathbb{Z}$$

eg. $4x + 8y = 6$

$$\begin{aligned} &4 \times 2 + 8 \times -2 \\ &= 8 - 16 = -8 \end{aligned}$$

$$\begin{aligned} &4 \times -6 + 8 \times 4 \\ &= -24 + 32 \\ &= 8 \end{aligned}$$

$$\begin{aligned} &4 \times 4 + 8 \times -1 \\ &= 16 - 8 = 8 \end{aligned}$$

$$\begin{aligned} &4 \times -4 + 8 \times 6 \\ &= -16 + 48 = 32 \end{aligned}$$

No integral solution

$$4(x+2y) = \textcircled{6}$$

$$\Rightarrow x + 2y = 3/2$$

$\uparrow \quad \uparrow \quad \uparrow$
Integer

$$\underline{a}x + \underline{b}y = \underline{c}$$

$$\gcd(a, b) = \underline{g}$$

$$\underline{g}(\underbrace{kx + my}_{\text{Integer}}) = \underline{c}$$

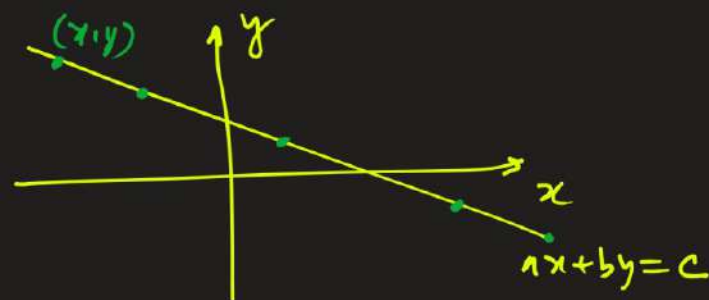
$\therefore c/g$ has to be an integer

$$\therefore \boxed{c \% g = 0}$$

$$ax + by = c$$

has integral solutions if and only if

c is divisible by $\gcd(a, b)$



$$\text{eg. } \underline{15}x + \underline{10}y = \underline{20}$$

$$\boxed{x=2, y=-1}$$

$$\boxed{x=4, y=-4}$$

$$\boxed{x=0, y=2}$$


```
#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
#define pb push_back
typedef vector<int> vi;
typedef vector<string> vs;
typedef long long int ll;
typedef unsigned long long int ull;
typedef vector<ll> vll;
typedef vector<ull> vull;

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

void solve() {
    int a,b,c,g;
    cin >> a >> b >> c;
    if(a>b) g = gcd(a,b);
    else g = gcd(b,a);
    if(c%g == 0) cout << "Yes" << endl;
    else cout << "No" << endl;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    cin >> t;
    FOR(i,1,t+1) {
        cout << "Case " << i << ": ";
        solve();
    }
    return 0;
}
```

-x-

$$ax + by = c$$

> iff c is div. by $\gcd(a, b)$ then there will be a solution.

Extended Euclidean Algorithm

$$\begin{aligned} ax + by &= c \rightarrow kx, ky & g = \gcd(a, b) \\ \underline{ax + by = g} &\rightarrow (x, y) \end{aligned}$$

$$c = k \cdot g$$

$$ax + by = kg$$

$$\text{eg. } 15x + 10y = 20 \quad 4x_0, 4y_0$$

$$15x + 10y = 5 \quad x_0, y_0$$

$$[1, -1] \checkmark$$

$$\begin{aligned} & \boxed{ax + by = g} \\ & \boxed{bx_0 + (a \div b) y_0 = g} \rightarrow x_0, y_0 \end{aligned}$$

$$\Rightarrow \underline{bx_0} + (a - \lfloor \frac{a}{b} \rfloor \cdot b) y_0 = g$$

$$\Rightarrow a(\underline{y_0}) + b(x_0 - \lfloor \frac{a}{b} \rfloor \cdot y_0) = g$$

$$\begin{aligned} x &= y_0 \\ y &= x_0 - \lfloor \frac{a}{b} \rfloor \cdot y_0 \end{aligned}$$

$$g(a, b) \rightarrow g(\underline{b}, \underline{a \% b})$$

$$\begin{aligned}
 & \underbrace{ax + by = g} \\
 & \quad \swarrow \quad \searrow \\
 & bx + \underbrace{(a \% b)y} = g \quad \swarrow \quad \searrow \quad \underline{x_0, y_0} \\
 & \quad \swarrow \quad \searrow \\
 & (a \% b)x + (b \% (a \% b))y = g \quad \swarrow \quad \searrow \quad \underline{x_1, y_1}
 \end{aligned}$$

$$\boxed{\underline{g, 0}} \quad \text{with} \quad \textcircled{1, k}$$

$$gx + by = g$$

$$\begin{aligned}
 ax + by &= g \\
 \underline{ax + by} &= \underline{c}
 \end{aligned}$$

$$\underline{ax} + \underline{by} = c$$

$$\boxed{\underline{by} + \underline{ax} = c}$$

$$2x + 4y = 8$$

$$x = 4, y = 0$$

$$3x + 6y = 7$$

No


```

#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
#define pb push_back
typedef vector<int> vi;
typedef vector<string> vs;
typedef long long int ll;
typedef unsigned long long int ull;
typedef vector<ll> vll;
typedef vector<ull> vull;

int ex_gcd(int a, int b, int &x, int &y){
    if(b==0) {
        x = 1;
        y = 0; // y can be any integer
        return a;
    }
    int x0, y0;
    int g = ex_gcd(b, a%b, x0, y0);
    x = y0;
    y = x0 - (a/b)*y0;
    return g;
}

void solve() {
    int a,b,c,g,x,y;
    cin >> a >> b >> c;
    if(a>b) g = ex_gcd(a,b,x,y);
    else g = ex_gcd(b,a,y,x);
    if(c%g == 0)
        cout << "x = " << (c/g)*x << ", y = " << (c/g)*y << endl;
    else cout << "No Solution exists!" << endl;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    cin >> t;
    FOR(i,1,t+1) {
        cout << "Case " << i << ": ";
        solve();
    }
    return 0;
}

```

```
def f_gcd(a, b):  
    # Returns x, y, g  
    if b==0:  
        return 1, 0, a  
    x0, y0, g = f_gcd(b, a%b)  
    return y0, x0-(a//b)*y0, g  
  
def solve(i):  
    a,b,c = map(int, input().split())  
    if a>b:  
        x,y,g = f_gcd(a,b)  
    else:  
        y,x,g = f_gcd(b,a)  
    if c%g == 0:  
        print(f'Case {i}: x = {(c//g)*x}, y = {(c//g)*y}')  
    else:  
        print(f'Case {i}: No Solution Exists!')  
  
t = int(input())  
for i in range(1, t+1):  
    solve(i)
```