Number Theory Lecture 3

Sunday, 7 July 2024 5:32 PM

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

gcd (12, 18) = ? (6)

2	12
2	6
3	3
	1

$12 = 2^2 \times 3^1$ ✓

$18 = 2^1 \times 3^2$ ✓

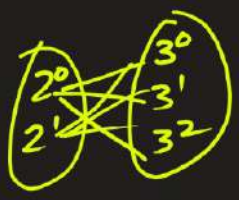
$\gcd(12,18) = 2^1 \times 3^1$

2	18
3	9
3	3
	1

0or1 0or1or2

$2^{-} \times 3^{-}$

#divisors of 18
$= 2 \times 3$
$= 6$



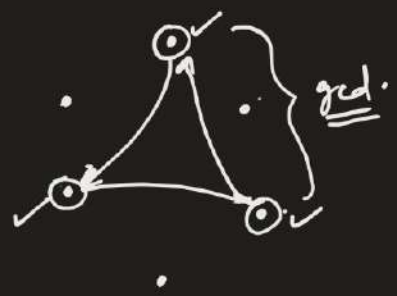
Divisors of 12 :-

$2^0 \times 3^0 = 1$
$2^1 \times 3^0 = 2$
$2^2 \times 3^0 = 4$
$2^0 \times 3^1 = 3$
$2^1 \times 3^1 = 6$
$2^2 \times 3^1 = 12$

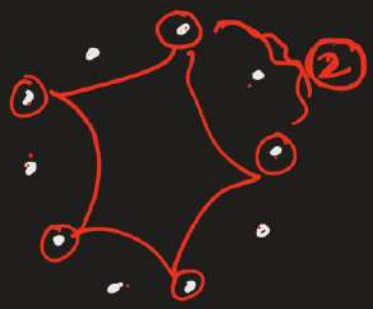
Divisors of 18 :-

$2^0 \times 3^0 = 1$
$2^0 \times 3^1 = 3$
$2^0 \times 3^2 = 9$
$2^1 \times 3^0 = 2$
$2^1 \times 3^1 = 6$
$2^1 \times 3^2 = 18$

gcd (4, 6)

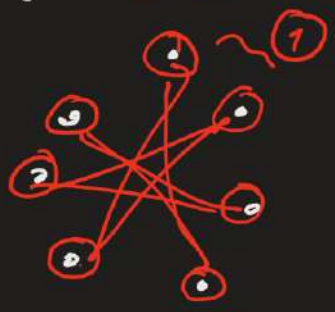


gcd (8, 10)



gcd (4, 8)

gcd (7, 3)



• gcd (a, 1) = 1

a > 0

- $gcd(a, 1) = 1$ $\quad a>0$
- $gcd(a, 0) = a$
- $gcd(a, b) \leq \min(a, b)$
- $gcd(a, a) = a$
- $gcd(a, b) = gcd(b, a)$

—×—

Q. How to find gcd of a and b?

$a \geq b$

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

$O(b)$

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

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

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

$\Rightarrow a | g$ and $b | g$

$\Rightarrow a-b | g$ $= \left(\frac{a}{g} - \frac{b}{g}\right)$

Euclidean Algorithm to find gcd

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

$a > b$

```
gcd(a, b) {
    if b == 0
        return a;
    {
        if (a-b > b)
            return gcd(a-b, b);
        else
            return gcd(b, a-b);
    }
}
```

$O(b)$

$Time = O(a)$

$a-b < 0$ if $b > a$

gcd(12, 10)
= gcd(10, 2)
= gcd(8, 2)
= gcd(6, 2)
= gcd(4, 2)
= gcd(2, 2)
= gcd(2, 0)
= 2

gcd(18, 12)
= gcd(12, 6)
= gcd(6, 6)
= gcd(6, 0)
= 6

gcd(8, 1)
= gcd(7, 1)
= gcd(6, 1)
= gcd(5, 1)
⋮
= gcd(1, 1)
= gcd(1, 0)
= 0

gcd(19, 11)
= gcd(11, 8)
= gcd(8, 3)
= gcd(5, 3)
= gcd(3, 2)
= gcd(2, 1)
= gcd(1, 1)
= gcd(1, 0)
= 1

Optimized Euclidian algorithm

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

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

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

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

$\vdots$

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

$gcd(16, 3)$

$= gcd(16-3, 3)$

$= gcd(16-3\times2, 3)$

$= gcd(16-3\times3, 3)$

$= gcd(16-3\times4, 3)$

$= gcd(16-3\times5, 3)$

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

$gcd(12, 8)$

$= gcd(8, 4)$

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

12

$gcd(12, 2)$

6

How many times can I subtract b from a

$\lfloor a/b \rfloor$ times

$gcd(12, 5)$

$= gcd(7, 5)$

$= gcd(5, 2)$

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

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

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

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

$gcd(16,3)$

$= gcd\left(16 - \left\lfloor \frac{16}{3} \right\rfloor \cdot 3,\ 3\right)$

$= gcd(16-15,\ 3)$

$= 1$

$16 \% 3$

$a \% b < b$

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

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

$a \geq b$

Time = O( )

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

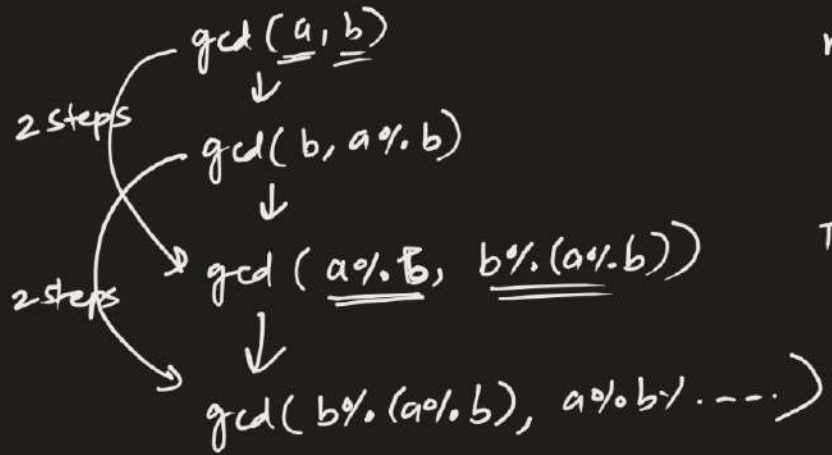
$a \% b \leq a/2$

$\min(a,b) = b$

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

$T(n) = O(\log n)$

$O(\log b)$

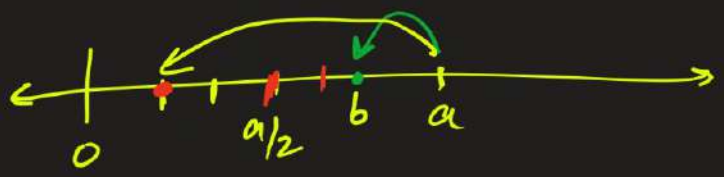


gcd(8,5)

gcd(b, a%b), ...

gcd(8,5)
= gcd(5,3)
= gcd(3,2)
=

$a\%b \leq a/2$



= x =

Linear Diophantine Equations

Find integral solution to the equation

$ax + by = c$

Integers

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

eg. $4x + 8y = 6$

$4 \times 2 + 8 \times -2$
$= 8 - 16 = -8$

$4 \times 4 + 8 \times -1$
$= 16 - 8 = 8$

$4 \times -4 + 8 \times 6$
$= -16 + 48 = 32$

$4 \times -6 + 8 \times 4$
$= -24 + 32$
$= 8$

No integral solution.

$y(x+2y) = 6$

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

Integer

$ax + by = c$

$\gcd(a, b) = g$

$g(kx + my) = c$

Integer

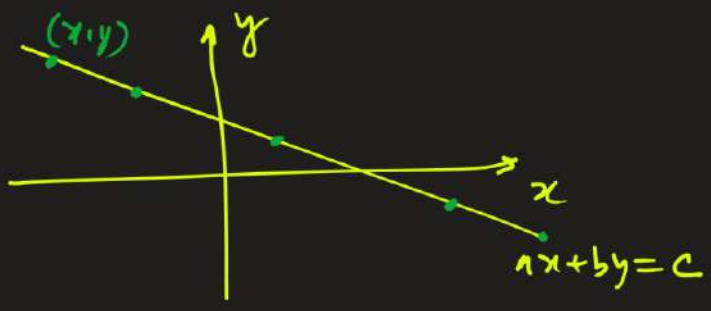
$\therefore c/g$ has to be an integer

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

$ax + by = c$

has integral solutions if and only if

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



eg: $15x + 10y = 20$

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

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

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

⋮

https://www.spoj.com/problems/CEQU/

```cpp
#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;
}
```

$ax + by = c$

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

Extended Euclidean Algorithm

$ax + by = c \rightarrow kx, ky$ $\quad g = \gcd(a,b)$ $\quad c = k \cdot g$

$ax + by = g \rightarrow (x, y)$

$ax + by = kg$

eg: $15x + 10y = 20 \rightarrow 4x_0, 4y_0$

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

$4, -4$

$ax + by = g$

$b x_0 + (a \% b)\, y_0 = g \rightarrow x_0, y_0$

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

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

$g(a,b)$

$g(b, a \% b)$

$x = y_0$

$y = x_0 - \lfloor \frac{a}{b} \rfloor \cdot y_0$

$ax + by = g$

$bx + (a\%b)y = g$ ← x_0, y_0

$(a\%b)x + (b\%(a\%b))y = g$ ← x_1, y_1

$\vdots$

$g, 0$ (1, k)

$gx + 0y = g$

$ax + by = g$

$ax + by = c$

$ax + by = c$

$by + ax = c$

$2x + 4y = 8$

$3x + 6y = 7$

$x = 4, y = 0$

No

```cpp
#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;
}
```

```python
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)
```