Competitive Programming

From Problem 2 Solution in O(1)

Number Theory

Extended Euclidean algorithm

Mostafa Saad Ibrahim
PhD Student @ Simon Fraser University

Finding GCD

- We already computed it using Euclid's algorithm $\gcd(a, 0) = a$ $\gcd(a,b) = \gcd(b, a \bmod b)$
- Remember aso: $a \bmod b = a - b \left\lfloor \frac{a}{b} \right\rfloor$
- gcd(a = 1180, b = 482)
  - 1180 = 482 * 2 + 216
  - = gcd(482, 1180%482 = 216) = 2
- Notice the equation:
  - 216 = 1180 - 482 * 2 = 1180 + 482 * (-2)
- Extended algorithm utilizes the quotient to compute following

$$ax + by = \gcd(a, b).$$

Finding GCD

GCD(A, B)	A % B	⌊A / B⌋	A % B = A - BQ = **A + B(-Q)**
(1180, 482)	216	2	216 = 1180 + 482 * (-2)
(482, 216)	50	2	50 = 482 + 216 (-2)
(216, 50)	16	4	16 = 216 + 50 (-4)
(50, 16)	2	3	2 = 50 + 16 (-3)
(16, 2)	0	8	0 = 16 + 2 (-8)
(2, 0)	gcd(A, B) = 2		

Extended GCD

GCD(A, B)	A % B = **A + B(-Q)**	Replace A%B with its equation
(1180, 482)	216 = 1180 + 482 * (-2)	2 = 482 (13)+ [1180 + 482 * (-2)] (-29) = **1180 (-29) + 482 * (71)**
(482, 216)	50 = 482 + 216 (-2)	2 = 216 (-3) + [482 + 216 (-2)] (13) = **482 (13)+ 216 (-29)**
(216, 50)	16 = 216 + 50 (-4)	2 = 50 + [216 + 50 (-4)] (-3) = **216 (-3) + 50 (13)**
(50, 16)	2 = 50 + 16 (-3)	2 = 50 + **16** (-3) => **Replace 16 with its equation**
(16, 2)	0 = 16 + 2 (-8)	
(2, 0)	gcd(A, B) = 2	

We can now write code to do this replacements from bottom to up...it will be annoying code
Let's make observations based on x and y

Extended GCD

GCD(A, B)	⌊A / B⌋	gcd(a, b) = a * x + b * y	x = prev_y	y = prev_x - q * x
(1180, 482)	2	2 = 1180 (**-29**) + 482 * (71)	-29	13 - 2 * (-29) = 71
(482, 216)	2	2 = 482 (13)+ 216 (-**29**)	**13**	-3 - 2 * 13 = -29
(216, 50)	4	2 = 216 (**-3**) + 50 (13)	-3	1 - 4 * (-3) = **13**
(50, 16)	3	2 = 50 + 16 (**-3**)	1	0 - 3 * 1 = -3
(16, 2)	8	2 = **0** * 16 + 1 * 2	0	1
(2, 0)		2 = 1 * 2 + **0** * 0	1	0 (base case)

Extended GCD: Code

```
// ax + by = g = gcd(a, b)
ll extended_euclid(ll a, ll b, ll &x, ll &y) {
    if (b == 0) {
        x = 1, y = 0;
        return a;
    }
    // swap b, a and swap their x, y
    ll g = extended_euclid(b, a % b, y, x);
    // now our x = previous y
    y -= (a / b) * x;

    return g;
}
```

Note: Wiki has other iterative approach...but that should be easier to code and prove

Your turn: Update code to handle if a < 0 or b < 0
Your turn: Extend of case we have multiple integers [e.g. ax + by + cz = gcd(a, b, c)]

Bézout's identity

- Assume $a > 0$ and $b > 0$
- $ax + by = g = gcd(a, b)$ => we know that
- Can we generate **further** solutions?
- Is following valid:
- $a(x+b) + b(y-a) = g$
  - Yes, we added ab - ab, so same equation
- $a(x+b/g) + b(y-a/g) = g$
- $a(x+kb/g) + b(y-ka/g) = g$
- With easy math, we can generate!

Bézout's identity

- When one pair of Bézout coefficients (x, y) has been computed, further **pairs** can be represented in the form

$$\begin{array}{llll} \vdots & & & \\ 12 \times -10 & + & 42 \times 3 & = 6 \\ 12 \times -3 & + & 42 \times 1 & = 6 \\ 12 \times 4 & + & 42 \times -1 & = 6 \\ 12 \times 11 & + & 42 \times -3 & = 6 \\ 12 \times 18 & + & 42 \times -5 & = 6 \\ \vdots & & & \end{array}$$

$$\left(x + k\frac{b}{\gcd(a,b)},\ y - k\frac{a}{\gcd(a,b)}\right),$$

Bézout's identity

- Among these pairs of Bézout coefficients, exactly two of them satisfy

$$|x| < \left| \frac{b}{\gcd(a,b)} \right| \quad \text{and} \quad |y| < \left| \frac{a}{\gcd(a,b)} \right|.$$

- The Extended Euclidean algorithm always produces one of these two minimal pairs.
  - Extended (a = 12, b = 42): (x = -3, y = 1): 12*-3+42*1 = 6
  - b / 6 = 7 a / 6 = 2
  - |x| = 3 < 7 |y| = 1 < 2

Bézout's identity

- Bézout's identity plays role in some problems when comes to proving them
- Read its proof on wiki or may be here
- Remember x + kb /g and y - ka/g are equations
- E.g. we can arrange them to force value on X
  - X’ = x+(k*b)/g > 0 => **-x*g/b < k**
  - Y’ = y-(k*a)/gcd > 0 => **y*gcd/a > k**

تم بحمد الله

علمكم الله ما ينفعكم

ونفعكم بما تعلمتم

وزادكم علماً

- UVA(718, 11768, 10104, 10090, 10633, 10673), SRM358-D1-2,