Integer Gcd Algorithms, Copyright Michael Monagan

January 19, 2021 2:17 PM

Let a and b are integers.
How can we compute $\gcd(a,b)$?
Maple! igcd(a,b);

$$\gcd(\underset{7\cdot 10}{70}, \underset{5\cdot 21}{105}) = 5\cdot 7 = 35.$$

Euclid's alg. ~300 BC.
Stein's alg. 1961.

Theorem (division). Let $a, b \in \mathbb{Z}$ with $\boxed{b>0}$. There exist unique integers q and r satisfying $a = bq + r$ and $\boxed{0 \le r < b}$.

quotient (q) remainder (r) $\Rightarrow$ uniqueness.

$23 \div 5$ $q=4, r=3$ $23 = 5\cdot 4 + 3$

Maple r := irem(a,b); r := irem(a,b,'q');
q := iquo(a,b); q := iquo(a,b,'r');

Note: if $r=0$ we say b divides a and we write $b|a$.
Note: if $b<0$ we require $0 \le r < |b|$ for uniqueness.

Definition (gcd) Let $a, b \in \mathbb{Z}$ not both 0, and $g \in \mathbb{Z}$.
g is a greatest common divisor of a and b written $\gcd(a,b)$ if

$\gcd(6,4) = \pm 2$

(i) $g|a$ and $g|b$ (common divisor)
(ii) if $h|a$ and $h|b$ then $h|g$ (greatest).

(iii) $g>0$ to impose uniqueness.

Lemma Let $a, b \in \mathbb{Z}$, $a>0, b>0$ and $a = bq + r$ with $0 \le r < b$.
Then (1) $\gcd(a,b) = \gcd(r,b)$
(2) $\gcd(a,b) = \gcd(a-b, b)$

Proof of (1). Let $g = \gcd(a,b)$ and $h = \gcd(r,b)$. $g=h$?
We will show $g|h$ and $h|g$. Since $g>0$ and $h>0 \Rightarrow g=h$.

[illegible]

We will show $g \mid h$ and $h \mid g$. Since g ...

$(g \mid h)$ $\quad \left.\begin{array}{l} g=\gcd(a,b) \overset{(i)}{\Rightarrow} g \mid a \text{ and } g \mid b \\ a=bq+r \Rightarrow r=a-bq \end{array}\right\} \Rightarrow \left.\begin{array}{l} g \mid r \\ g \mid b \end{array}\right\} \underset{(ii)}{\Rightarrow} g \mid \gcd(r,b) = h.$

$(h \mid g)$ Exercise.

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

Euclid used (2). $\quad \gcd(a,b) = \gcd(a-b, b).$

9 = remainder of $39 \div 15$.

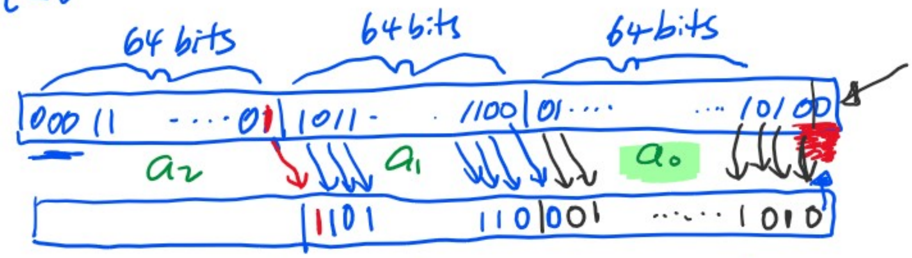
$\gcd(39,15) = \gcd(24,15) = \gcd(9,15)$
$= \gcd(15,9) = \gcd(6,9) = \gcd(9,6) = \gcd(3,6)$
$= \gcd(6,3) = \gcd(3,3) = \gcd(0,3) = 3.$

Is (2) faster than (1)? $\quad \gcd(2 \cdot 10^6, 2) = 2 \quad$ NO.

Binary GCD Algorithm (J. Stein 1961).

Suppose $a = \sum_{i=0}^{n-1} a_i B^i$ where $B = 2^k$ e.g. $B = 2^{64}$

$a =$



Observations

1. Easy to test if $2 \mid a$. $\quad$ `if( a[0] & 1 == 0)` $\quad O(1)$.
2. Easy to divide by 2. Shift a right $\quad O(n)$.

(1) $\gcd(0,n) = n$, $\gcd(a,b) = \gcd(b,a)$

(2) $\gcd(2m, 2 \cdot n) = 2 \cdot \gcd(m,n)$

(3) $\gcd(\cancel{2}m, 2n+1) = \gcd(m, 2n+1)$

(4) $\gcd(2m+1, 2n+1) = \gcd((2m+1)-(2n+1), 2n+1) = \gcd(m-n, 2n+1).$

$2m+1 \geq 2n+1$ Euc $\quad = 2(m-n)$

$$\gcd(66,36) \underset{(2)}{=} 2\cdot\gcd(33,18) \underset{(3)}{=} 2\cdot\gcd(33,9) \underset{(4)}{=} 2\left[\gcd(24,9) = \gcd(12,9)\right]$$

$$= 2\cdot\gcd(6,9) = 2\cdot\gcd(3,9) = 2\gcd(9,3)$$

$$\underset{(4)}{=} 2\gcd(6,3) = 2\gcd(3,3) \underset{(4)}{=} 2\gcd(0,3) = 2\cdot 3 = 6.$$

The only divisions are by 2 which is easy in binary. $B=2^k$.

Algorithm BINGCD Assume $0 < A, B < 2^n$.
Input: $A, B \in \mathbb{Z}^+$ Output $g = \gcd(A,B)$.

Set $k=0$, $a \leftarrow A$, $b \leftarrow B$.

Loop: if $a<b$ then interchange a and b — $O(n)$.

- CASE $\underline{b \text{ even}}$ a odd : $b \leftarrow b/2$ — $O(n)$
- CASE a even b odd : $a \leftarrow a/2$ — $O(n)$
- CASE a even b even
  $k \leftarrow k+1$; $a \leftarrow a/2$; $b \leftarrow b/2$; — $2O(n)$.
- CASE a odd b odd. $a \leftarrow (a-b)/2$ — $2O(n)$.

if $a=0$ then $g \leftarrow 2^k \cdot b$; return g; — $\leq O(n)$.
shift b left by k bits

goto LOOP.

---

while true do . . . . return g; - - - - od;

Let $T(n)$ be the cost of alg. BINGCD.
Suppose $0 < A, B < 2^n$ (n bits long).
How many times is the loop executed? $\leq 2n$.
because one of a and/or b gets smaller by ≥ 1 bit.

$T(n) \leq 2n\,(O(n) + O(1) + 2O(n)) + O(n)$

because one of a and/or b gets smaller by ≥ 1 ...

$$T(n) \leq 2n\Big(\underset{\text{if } a<b}{O(n)} + \underset{a/b \text{ even/odd}}{O(1)} + 2O(n)\Big) + \underset{2^k \cdot b}{O(n)}$$

$$= 2n\big(O(n+1+2n)\big) + O(n)$$

$$= 2n\big(O(3n+1) = O(n)\big) + O(n) = O(2n^2) + O(n)$$

$$= O(3n^2) = \underline{\underline{O(n^2)}}.$$

For $0 < A, B < 2^n$

Euclid's algorithm is $O(n^2)$ Knuth Vol II

Stein's algorithm is $O(n^2)$

Schönhage & Strassen is $O(M(n) \log n)$ where $M(n)$ is the cost of multiplying $A \times B$.
1971

Is $\gcd \in O(M(n))$? Open.