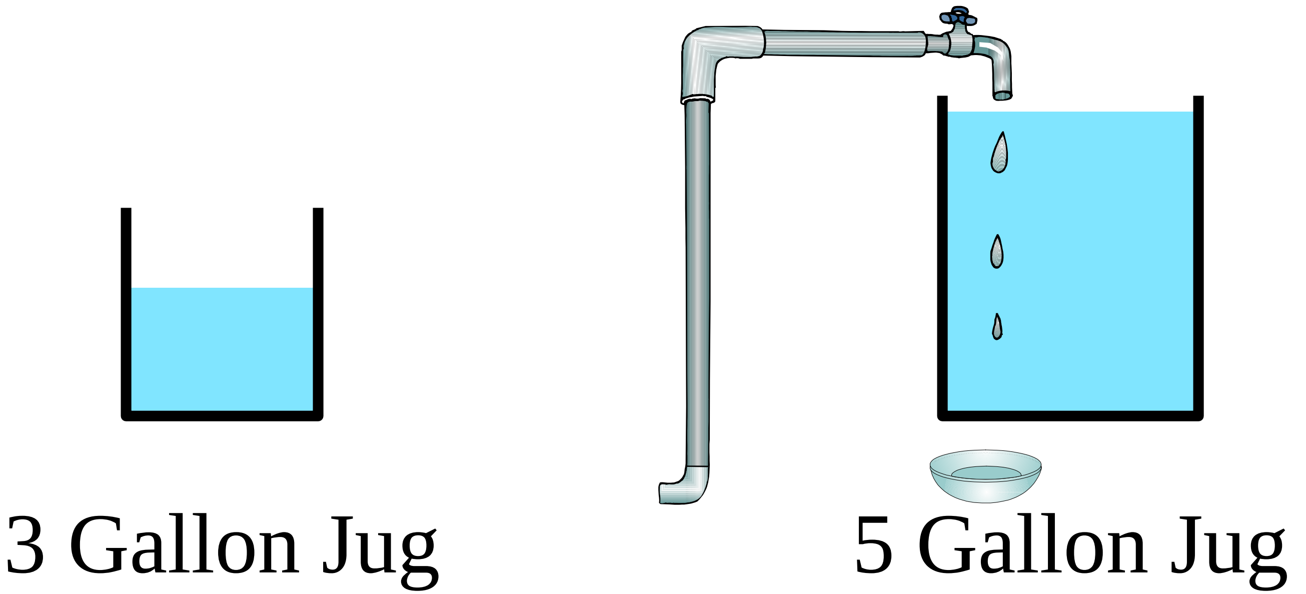
Greatest Common Divisors



Common Divisors

c is a *common divisor* of a and b means $c|a$ and $c|b$.

$gcd(a,b)$ ::= the **greatest common divisor** of a and b.

Say $a=8$, $b=10$, then 1,2 are common divisors, and $gcd(8,10)=2$.

Say $a=10$, $b=30$, then 1,2,5,10 are common divisors, and $gcd(10,30)=10$.

Say $a=3$, $b=11$, then the only common divisor is 1, and $gcd(3,11)=1$.

Claim. If p is prime, and p does not divide a, then $gcd(p,a) = 1$.

The Quotient-Remainder Theorem

For $b > 0$ and any a, there are *unique* integers

q ::= quotient(a,b), r ::= remainder(a,b), such that

$$a = qb + r \quad \textbf{and} \quad 0 \le r < b.$$

We also say **q = a div b** and **r = a mod b**.

When b=2, there is a unique q such that

a=2q or a=2q+1.

$$q = \lfloor \frac{a}{2} \rfloor$$

When b=3, there is a unique q such that

a=3q or a=3q+1 or a=3q+2.

$$q = \lfloor \frac{a}{3} \rfloor$$

Floor function $\lfloor x \rfloor$ **:** the greatest integer that is $\le x$

The Quotient-Remainder Theorem

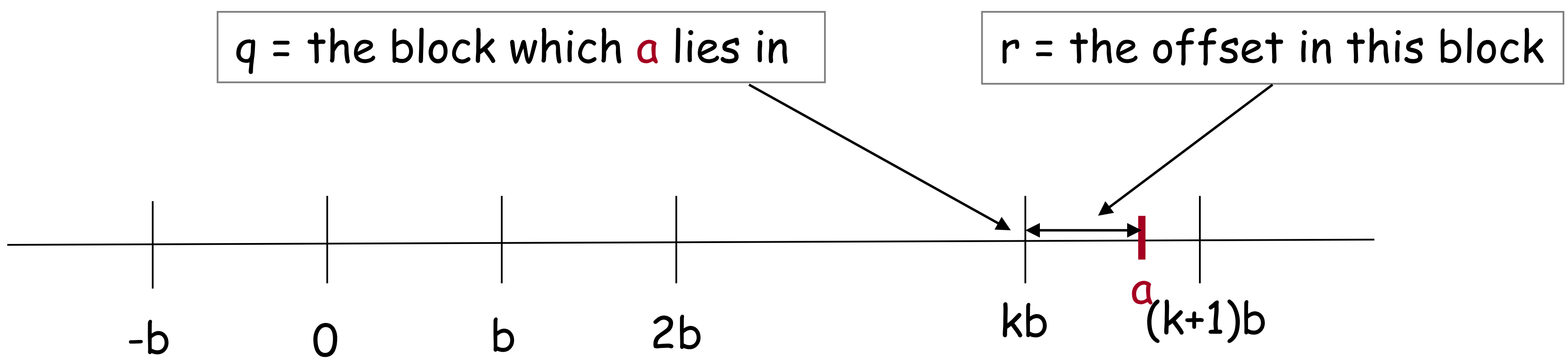
For b > 0 and any a, there are *unique* integers

q ::= quotient(a,b), r ::= remainder(a,b), such that

$$a = qb + r \quad \text{and} \quad 0 \leq r < b.$$

Given any b, we can partition the integers into blocks of b numbers.

For any a, there is a unique "position" for this number.



Clearly, given a and b, the numbers q and r are uniquely determined.

Greatest Common Divisors

Given a and b, how to compute gcd(a,b)?

Maybe try every number? Not easy for large numbers…
Do we have a better way to do it?

Let's say $a \geq b$.

1. If $a=kb$, then $gcd(a,b)=b$, and we are done.
2. Otherwise, by the Division Theorem, $a = qb + r$ where $r>0$.

Greatest Common Divisors

Let's say $a \geq b$.

1. If $a=kb$, then $\gcd(a,b)=b$, and we are done.
2. Otherwise, by the Division Theorem, $a = qb + r$ where $r>0$.

$a=12$, $b=8$ => $12 = 8 + 4$	$\gcd(12,8) = 4$	$\gcd(8,4) = 4$
$a=21$, $b=9$ => $21 = 2 \times 9 + 3$	$\gcd(21,9) = 3$	$\gcd(9,3) = 3$
$a=99$, $b=27$ => $99 = 3 \times 27 + 18$	$\gcd(99,27) = 9$	$\gcd(27,18) = 9$

Euclid: $\gcd(a,b) = \gcd(b,r)$!

Euclid's GCD Algorithm

a = qb + r

Euclid: gcd(a,b) = gcd(b,r)!

Assumption: a > b ≥ 0.

gcd(a,b)

if b = 0, then answer = a.

else

write a = qb + r $\leftarrow$ $q = \lfloor \frac{a}{b} \rfloor \qquad r = a - qb$

answer = gcd(b,r)

Example 1

```
gcd(a,b)
if b = 0, then answer = a.
else
  write a = qb + r
  answer = gcd(b,r)
```

GCD(102, 70)	102 = 70 + 32
= GCD(70, 32)	70 = 2x32 + 6
= GCD(32, 6)	32 = 5x6 + 2
= GCD(6, 2)	6 = 3x2 + 0
= GCD(2, 0)	

Return value: 2.

Example 2

```
gcd(a,b)
if b = 0, then answer = a.
else
  write a = qb + r
  answer = gcd(b,r)
```

GCD(252, 189)	$252 = 1 \times 189 + 63$
= GCD(189, 63)	$189 = 3 \times 63 + 0$
= GCD(63, 0)	

Return value: 63.

Example 3

```
gcd(a,b)
if b = 0, then answer = a.
else
  write a = qb + r
  answer = gcd(b,r)
```

GCD(662, 414)	662 = 1x414 + 248
= GCD(414, 248)	414 = 1x248 + 166
= GCD(248, 166)	248 = 1x166 + 82
= GCD(166, 82)	166 = 2x82 + 2
= GCD(82, 2)	82 = 41x2 + 0
= GCD(2, 0)	

Return value: 2.

Correctness of Euclid's GCD Algorithm

$$\boxed{a = qb + r}$$

$$\boxed{\text{Euclid: } gcd(a,b) = gcd(b,r)}$$

When r = 0:

Then $a = qb$, so $gcd(a, b) = b$;

$r = 0$, so $gcd(b, r) = gcd(b, 0) = b$.

Therefore, $gcd(a,b) = gcd(b,r)$.

Correctness of Euclid's GCD Algorithm

$a = qb + r$

Euclid: $gcd(a,b) = gcd(b,r)$

When r > 0:

Let d be a common divisor of b, r

$\Rightarrow b = k_1 d$ and $r = k_2 d$ for some k_1, k_2.

$\Rightarrow a = qb + r = qk_1 d + k_2 d = (qk_1 + k_2)d$ => d is a common divisor of a, b

Let d be a common divisor of a, b

$\Rightarrow a = k_3 d$ and $b = k_1 d$ for some k_1, k_3.

$\Rightarrow r = a - qb = k_3 d - qk_1 d = (k_3 - qk_1)d$ => d is a common divisor of b, r

So, {common factors of a, b} = {common factors of b, r}

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

Is Euclid's GCD Algorithm fast?

Naive algorithm: try every number.

Assumption: $a > b \geq 0$.

gcd(a,b)

Let d=1

1. If d|a and d|b, then store d.
2. Let d=d+1
3. If $d \leq b$, return to 1.

   else the answer = product of all stored "d"s

So the running time is about b iterations.

Is Euclid's GCD Algorithm fast?

Euclid's algorithm:

In two iterations, a, b are decreased by half. (why?)

$a = bq + r \geq b + r > 2r$

$\Rightarrow gcd(a,b) = gcd(b,r)$ where $r < a/2$

Similarly, if $b = rq' + r'$, then

$gcd(b,r) = gcd(r,r')$ where $r' < b/2$

Suppose the algorithm stops in 2k iterations.

Then $2^d \geq 2^{k-1}$. (Suppose $2^{d+1} \geq b > 2^d$)

So the running time is about $2\log_2 b$ iterations.

Exponentially faster!!

Linear Combination vs Common Divisor

Greatest common divisor

d is a common divisor of a and b if $d|a$ and $d|b$

gcd(a,b) = greatest common divisor of a and b

Smallest positive integer linear combination

d is an **integer linear combination** of a and b if $d=sa+tb$ for integers s,t.

spc(a,b) = **smallest** **positive** integer linear **combination** of a and b

Theorem. gcd(a,b) = spc(a,b)

Linear Combination vs Common Divisor

Theorem. gcd(a,b) = spc(a,b)

For example, the *greatest* common divisor of 52 and 44 is 4.
And 4 is an integer linear combination of 52 and 44:

$$6 \cdot 52 + (-7) \cdot 44 = 4$$

Furthermore, no integer linear combination of 52 and 44 is equal to a smaller positive integer.

To prove the theorem, we will prove:

gcd(a,b) ≤ spc(a,b)

gcd(a,b) | spc(a,b)

gcd(a,b) ≥ spc(a,b)

spc(a,b) divides a and b

GCD $\leq$ SPC

Claim. If $d \mid a$ and $d \mid b$, then $d \mid sa + tb$ for any s,t.

Proof.

$d \mid a \Rightarrow a = dk_1$

$d \mid b \Rightarrow b = dk_2$

$sa + tb = sdk_1 + tdk_2 = d(sk_1 + tk_2)$

$\Rightarrow d \mid (sa+tb)$

GCD | SPC

Let $d = \gcd(a,b)$. By definition, $d \mid a$ and $d \mid b$.

Let $f = \mathrm{spc}(a,b) = sa+tb$

According to the claim, $d \mid f$. So $\gcd(a,b) \leq \mathrm{spc}(a,b)$.

GCD $\geq$ SPC

We will prove that spc(a,b) is actually a common divisor of a and b.

First, show that spc(a,b) | a.

1. By the Division Theorem (since a $\geq$ spc(a,b)),

   $$a = q \times spc(a,b) + r \qquad \text{and} \qquad spc(a,b) > r \geq 0$$

2. Let spc(a,b) = sa + tb.
3. Then r = a – q x spc(a,b) = a – q x (sa + tb) = (1-qs)a + qtb.
4. So r is an integer linear combination of a and b with spc(a,b) > r.
5. This is only possible when r = 0.

Similarly, spa(a,b) | b.

Thus, spc(a,b) divides both a and b, which follows spc(a,b) $\leq$ gcd(a,b).

Application of the Theorem

Theorem. $\gcd(a,b) = \mathrm{spc}(a,b)$

Lemma. If $\gcd(a,b)=1$ and $\gcd(a,c)=1$, then $\gcd(a,bc)=1$.

By the **Theorem**, there exist s,t,u,v such that

$$sa + tb = 1$$
$$ua + vc = 1$$

So $(sa + tb)(ua + vc) = 1$

Expanding LHS gives

$$saua + savc + tbua + tbvc = 1$$

$\Rightarrow (sau + svc + tbu)a + (tv)bc = 1$

This implies $\mathrm{spc}(a,bc)=1$. By **Theorem**, we have $\gcd(a,bc)=1$.

Prime Divisibility

Theorem. $\gcd(a,b) = \mathrm{spc}(a,b)$

Lemma. p prime and $p|ab$ implies $p|a$ or $p|b$.

proof. W.l.o.g, assume p does not divide a. Then $\gcd(p,a)=1$.

So by **Theorem**, there exist s and t such that

$$sa + tp = 1$$

$$\underbrace{(sa)b}_{p|ab} + \underbrace{(tp)b}_{p|p} = b$$

$\boxed{\text{Hence } p|b}$

Corollary. If p is prime, and $p|\ a_1 \cdot a_2 \cdots a_m$ then $p|a_i$ for some i.

Fundamental Theorem of Arithmetic

Every integer $n>1$ has a *unique* factorization into primes:

$$p_0 \leq p_1 \leq \cdots \leq p_k$$

$$n = p_0\, p_1 \cdots p_k$$

Example:

61394323221 = 3·3·3·7·11·11·37·37·37·53

Unique Factorization

Theorem. There is a unique factorization.

Proof. Suppose there is a number with two different factorizations.

By Well-Ordering Principle, we choose the smallest such $n > 1$:

$$n = p_1 \cdot p_2 \cdots p_k = q_1 \cdot q_2 \cdots q_m$$

Since n is smallest, we must have that $p_i \neq q_j$ all i,j

(Otherwise, we can obtain a smaller counterexample.)

Since $p_1 | n = q_1 \cdot q_2 \cdots q_m$, so by Corollary $p_1 | q_i$ for some i.

Since both p_1, q_i are prime numbers, we must have $p_1 = q_i$.

contradiction!

Extended GCD Algorithm

How can we write gcd(a,b) as an integer linear combination?

This can be done by extending the Euclidean algorithm.

Example: $a = 259$, $b=70$

$259 = 3 \cdot 70 + 49$ $\qquad$ $49 = a - 3b$

$70 = 1 \cdot 49 + 21$ $\qquad$ $21 = 70 - 49$

$21 = b - (a-3b) = -a+4b$

$49 = 2 \cdot 21 + 7$ $\qquad$ $7 = 49 - 2 \cdot 21$

$7 = (a-3b) - 2(-a+4b) = \underline{3a - 11b}$

$21 = 7 \cdot 3 + 0$ $\qquad$ done, gcd = 7

Extended GCD Algorithm

Example: $a = 899$, $b=493$

$899 = 1\cdot 493 + 406$ so $406 = a - b$

$493 = 1\cdot 406 + 87$ so $87 = 493 - 406$

$= b - (a-b) = -a + 2b$

$406 = 4\cdot 87 + 58$ so $58 = 406 - 4\cdot 87$

$= (a-b) - 4(-a+2b) = 5a - 9b$

$87 = 1\cdot 58 + 29$ so $29 = 87 - 1\cdot 58$

$= (-a+2b) - (5a-9b) = \underline{-6a + 11b}$

$58 = 2\cdot 29 + 0$ done, gcd = 29

Die Hard

Simon says: On the fountain, there are 2 jugs, one is 5-gallon and the other is 3-gallon. Fill one with exactly 4 gallons of water and place it on the scale then the timer will stop. You must be precise; one ounce more or less will result in detonation. If you're still alive in 5 minutes, we'll speak.

Die Hard

Bruce: Wait, wait a second. I don't get it. Do you get it?

Samuel: No.

Bruce: Get the jugs. Obviously, we can't fill the 3-gallon jug with 4 gallons of water.

Samuel: Obviously.

Bruce: All right. I know, here we go. We fill the 3-gallon jug exactly to the top, right?

Samuel: Uh-huh.

Bruce: Okay, now we pour this 3 gallons into the 5-gallon jug, giving us exactly 3 gallons in the 5-gallon jug, right?

Samuel: Right, then what?

Bruce: All right. We take the 3-gallon jug and fill it a third of the way...

Samuel: No! He said, "Be precise." Exactly 4 gallons.

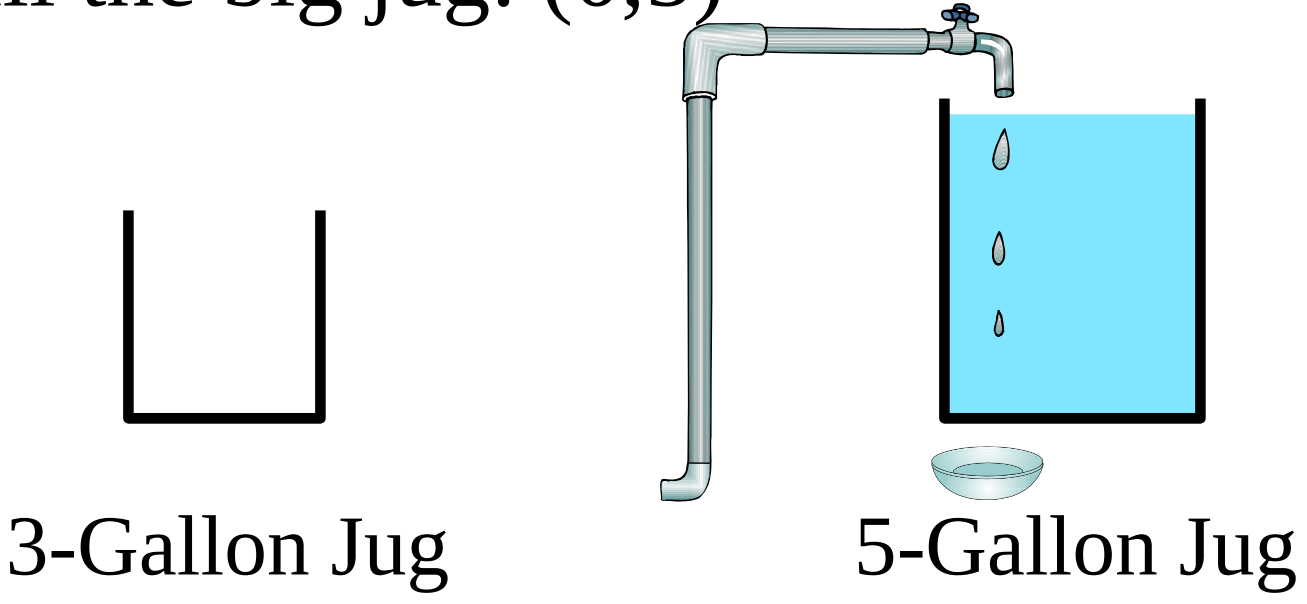
Bruce: Sh - -. Every cop within 50 miles is running his a** off and I'm out here playing kids games in the park.

Samuel: Hey, you want to focus on the problem at hand?

Die Hard

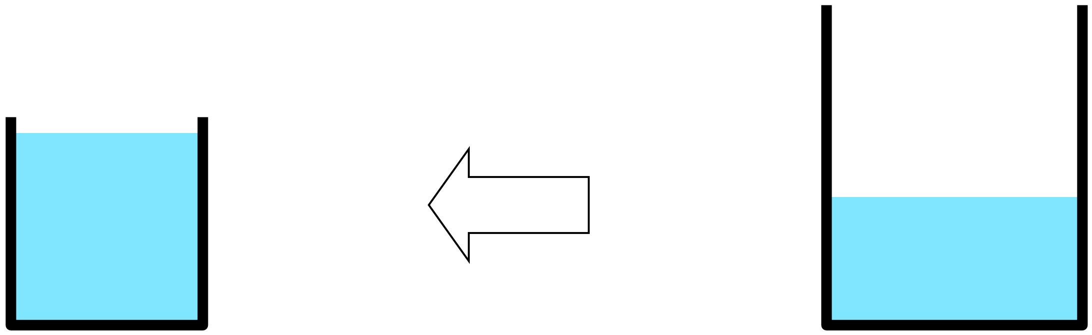
Start with empty jugs: (0,0)

Fill the big jug: (0,5)



Die Hard

Pour from big to little: (3,2)



3-Gallon Jug

5-Gallon Jug

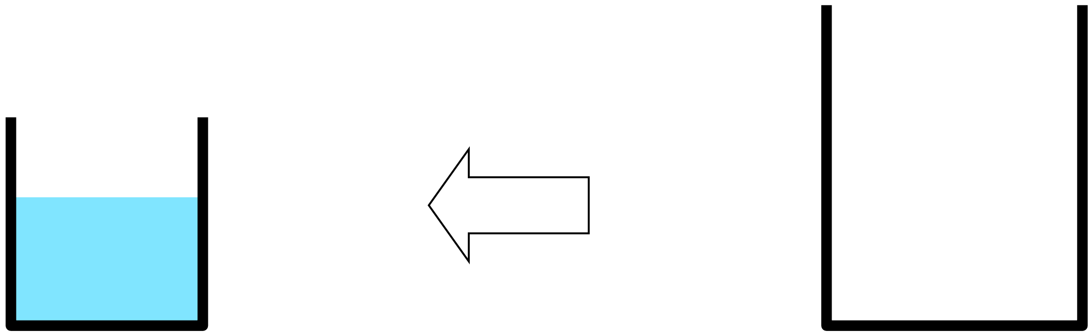
Die Hard

Empty the little: (0,2)

3-Gallon Jug   5-Gallon Jug

Die Hard

Pour from big to little: (2,0)

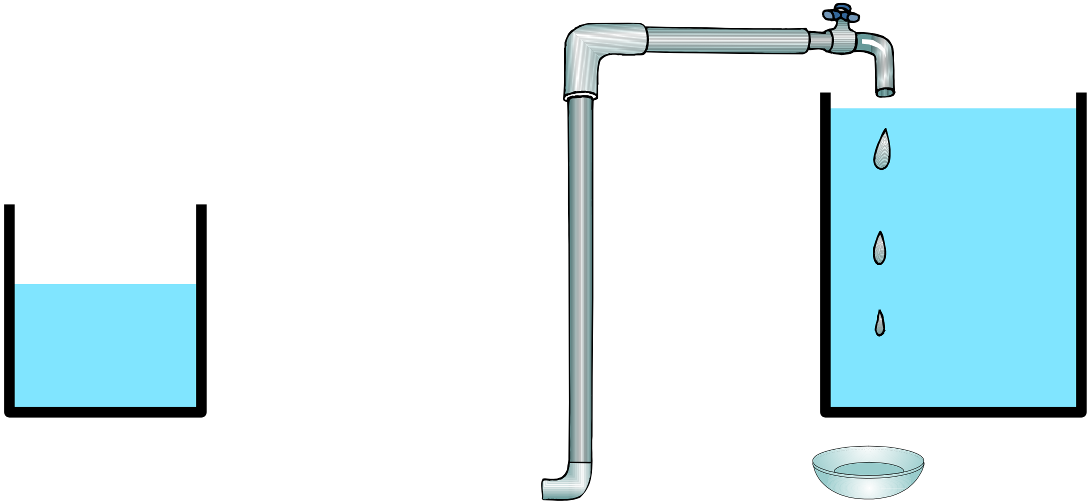


3-Gallon Jug

5-Gallon Jug

Die Hard

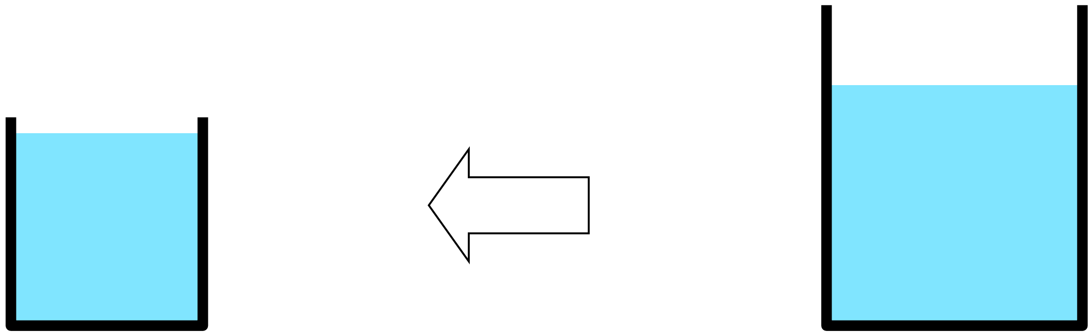
Fill the big jug: (2,5)



3-Gallon Jug 5-Gallon Jug

Die Hard

Pour from big to little: (3,4)

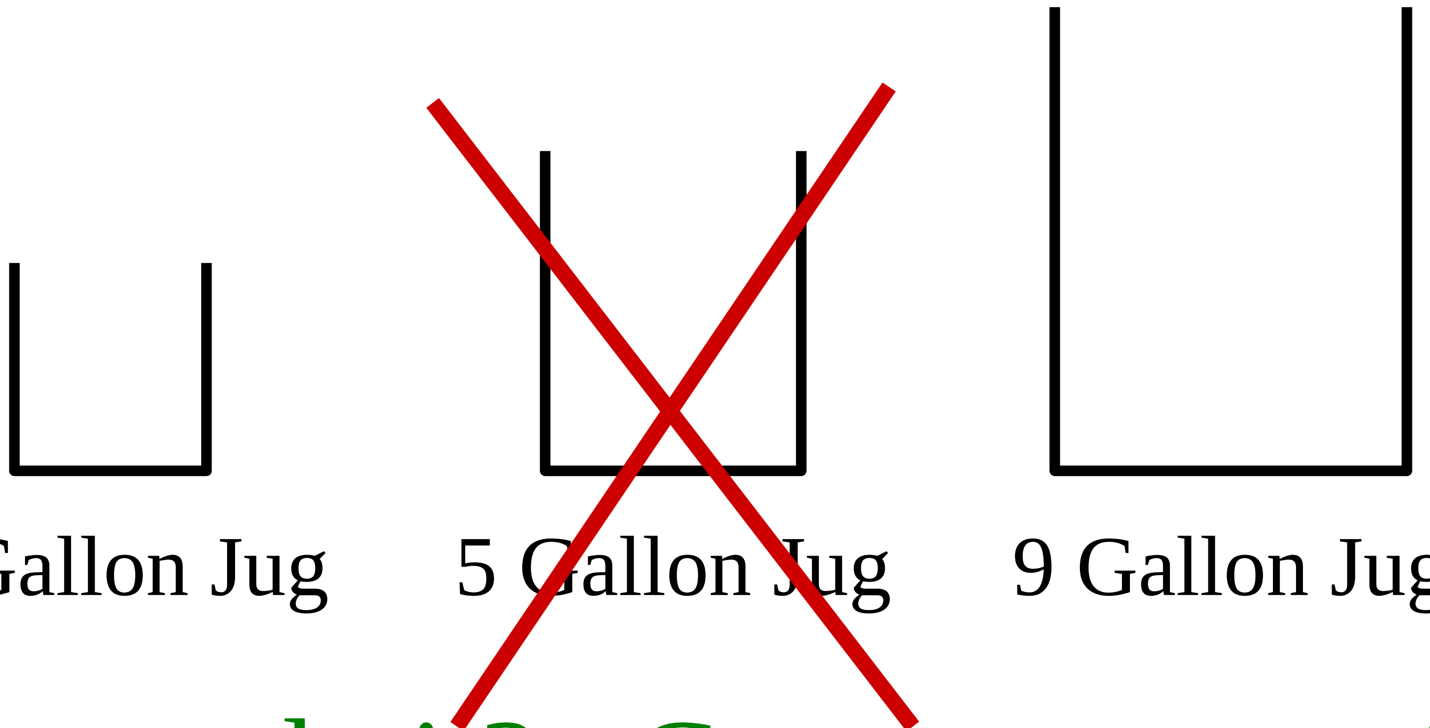


3-Gallon Jug 5-Gallon Jug

Done!!

Die Hard

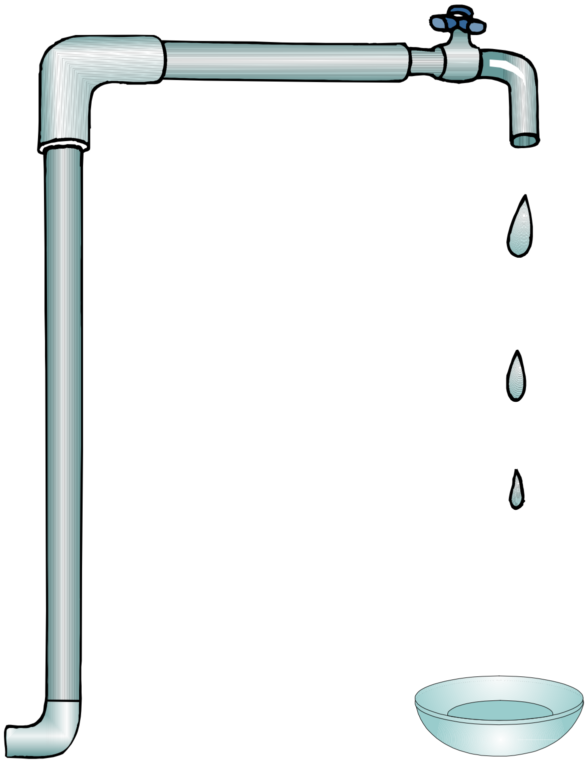
What if you have a 9 gallon jug instead?



Can you do it? Can you prove it?

Die Hard

Supplies:



Water

3-Gallon Jug

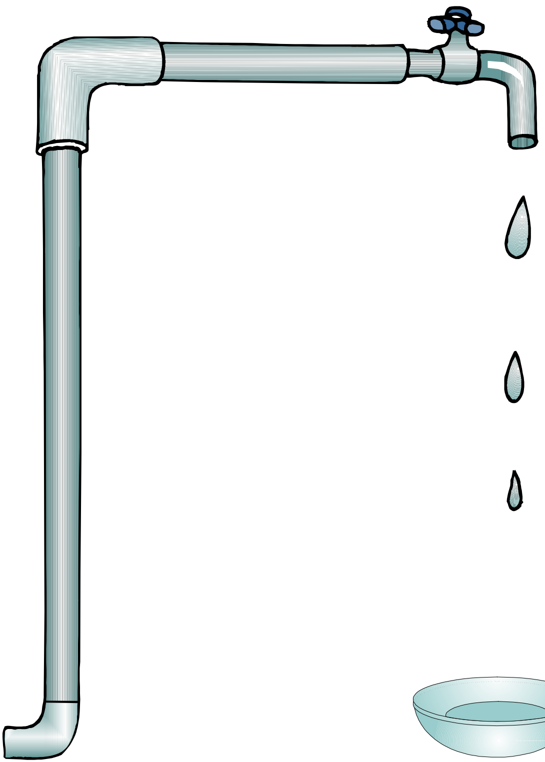
9-Gallon Jug

Invariant Method

Invariant: the number of gallons in each jug is a multiple of 3.

i.e., $3|L$ and $3|B$ (3 divides both L and B)

Corollary. It is impossible to have exactly 4 gallons in one jug.



Bruce Dies!

Generalized Die Hard

Can Bruce form 3 gallons using 21 and 26-gallon jugs?

This question is not so easy to answer without number theory.

General Solution for Die Hard

Invariant in Die Hard Transition:

Suppose that we have water jugs with capacities B and L.

Then the amount of water in each jug is always an integer linear combination of B and L.

Lemma. gcd(a, b) divides any integer linear combination of a and b.

Let d = gcd(a,b). Then

$$d|a \quad \text{and} \quad d|b$$

So $d|ax+by$.

Corollary. The amount of water in each jug is a multiple of gcd(a,b).

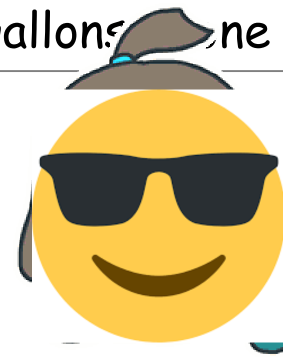
General Solution for Die Hard

Corollary. The amount of water in each jug is a multiple of gcd(a,b).

Given jug of 3 and jug of 9, is it possible to have exactly 4 gallons in one jug?

NO, because gcd(3,9)=3, and 4 is not a multiple of 3.

Given jug of 21 and jug of 26, is it possible to have exactly 3 gallons in one jug?



gcd(21,26)=1, and 3 is a multiple of 1,

so this means possible??

Theorem. Given water jugs of capacity a and b with a ≤ b,
it is possible to have exactly k (≤ b) gallons in one jug
if and only if k is a multiple of gcd(a,b).

General Solution for Die Hard

Theorem. Given water jugs of capacity a and b with $a \le b$, it is possible to have exactly k ($\le b$) gallons in one jug if and only if k is a multiple of gcd(a,b).

Given jug of 21 and jug of 26, is it possible to have exactly 3 gallons in one jug?

$$\gcd(21,26) = 1$$
$$\Rightarrow 5 \times 21 - 4 \times 26 = 1$$
$$\Rightarrow 15 \times 21 - 12 \times 26 = 3$$

Repeat 15 times:

1. Fill the 21-gallon jug.
2. Pour all the water in the 21-gallon jug into the 26-gallon jug.
   Whenever the 26-gallon jug becomes full, empty it out.

General Solution for Die Hard

$15 \times 21 - 12 \times 26 = 3$

Repeat 15 times:

1. Fill the 21-gallon jug.
2. Pour all the water in the 21-gallon jug into the 26-gallon jug.

   Whenever the 26-gallon jug becomes full, empty it out.

Claim. There must be exactly 3 gallons left after this process.

1. Totally we have filled 15x21 gallons.
2. We pour out t multiple of 26 gallons.
3. The 26 gallon jug can only hold the volume between 0 and 26.
4. So t must be 12.
5. And there is exactly 3 gallons left.

General Solution for Die Hard

Given two jugs with capacity A and B with $A \leq B$, the target is C.

If $\gcd(A,B)$ does not divide C, then it is impossible.

Otherwise, compute $C = sA + tB$. (We can always make $s > 0$.)

Repeat s times:

1. Fill the A-gallon jug.
2. Pour all the water in the A-gallon jug into the B-gallon jug.
   Whenever the B-gallon jug becomes full, empty it out.

The B-gallon jug will be emptied exactly t times.

After that, there will be exactly C gallons in the B-gallon jug.