Binary GCD algorithm in C
Source: http://en.wikipedia.org/wiki/Binary_GCD_algorithm
This code is licensed under the Creative Commons Attribution-ShareAlike License.
It is from the Wikipedia article "Counting sort" dated 2010-01-17.
The binary GCD algorithm, also known as Stein's algorithm, is an
algorithm that computes the greatest common divisor of two nonnegative
integers. It gains a measure of efficiency over the ancient Euclidean
algorithm by replacing divisions and multiplications with shifts, which
are cheaper when operating on the binary representation used by modern
computers.
Following is an implementation of the algorithm in C, taking two
(non-negative) integer arguments u and v. It first removes all common
factors of 2 using identity 2, then computes the GCD of the remaining
numbers using identities 3 and 4, and combines these to form the final
answer.

typedef unsigned
long long uint64;

uint64 gcd(uint64
u, uint64 v)

int shift;

GCD(0,x) := x

«DecisionInput»
u == 0 || v == 0

Let shift := lg K, where K is the greatest power of 2
dividing both u and v.

[No]

[Yes]

shift = 0

«DecisionInput»
((u | v) & 1) == 0

return u | v;

«DecisionInput»
(u & 1) == 0

[No]

[Yes]

u >>= 1;
v >>= 1;

[No]

[Yes]

From here on, u
is always odd.

u >>= 1;

++shift

Loop X

«DecisionInput»
(v & 1) == 0

[Yes]

[No]

[Yes]

v >>= 1;

Now u and v are both odd, so diff(u, v) is even.
Let u = min(u, v), v = diff(u, v)/2.

«DecisionInput»
u < v

[Yes]

[No]

v -= u;

uint64 diff = u - v;
u = v;
v = diff;

v >>= 1;

«DecisionInput»
v != 0

[No]

return u << shift;