If $d|a$ and $d|b$ and $d>0$, then $\left(\frac{a}{d}, \frac{b}{d}\right) = \frac{1}{d}(a,b)$

If $(a,b) = g$, then $\left(\frac{a}{g}, \frac{b}{g}\right) = 1$.

If $(a,m) = (b,m) = 1$, then $(ab,m) = 1$.

let $ax_1 + my_1 = 1 \Rightarrow b = abx_1 + mby_1$

$bx_2 + my_2 = 1 \Rightarrow (abx_1 + mby_1)x_2 + my_2 = 1$

$\Rightarrow ab(x_1x_2) + m(by_1x_2 + y_2) = 1 \Rightarrow \underline{(ab,m) = 1}$.

We say that a and b are relatively prime in case $(a,b)=1$ and that $a_1, a_2, \ldots, a_n$ are relatively prime in case $(a_1, a_2, \ldots, a_n) = 1$. We say that $a_1, a_2, \ldots, a_n$ are relatively prime in pairs in case $(a_i, a_j) = 1$ for all $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$ with $i \neq j$.

The fact that $(a,b) = 1$ is sometimes expressed by saying that a and b are coprime, or by saying that a is prime to b.

For any integer x, $(a,b) = (b,a) = (a,-b) = (a, b+ax)$.

$(a,b) = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + by'|\} = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|by' + ax'|\} = (b,a)$

$(a,b) = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + by'|\} = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + (-b)(-y')|\}$

$= \underset{x',y'' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + (-b)y''|\} = (a,-b)$.

$(a, b+ax) = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + (b+ax)y'|\} = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|a\underline{(x' + xy')} + by'|\}$

$\Rightarrow \underline{(a, b+ax) \geq (a,b)}$

$(a,b) = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|ax' + by'|\} = \underset{x',y' \in \mathbb{Z}}{\operatorname{Min}} \{|a\underline{(x' - xy')} + (b+ax)y'|\}$

$\Rightarrow \underline{(a,b) \geq (a, b+ax)} \quad \Rightarrow \underline{(a,b) = (a, b+ax)}$

If $c \mid ab$ and $(b,c) = 1$, then $c \mid a$.

$bx + cy = 1 \Rightarrow a = abx + acy$.

$c \mid ab$ and $c \mid ac \Rightarrow c \mid (abx + acy)$

$\Rightarrow$ $\underline{c \mid a}$.

<u>The Euclidean algorithm</u>: Given integers b and $c > 0$, we make a repeated application of the division algorithm to obtain a series of equations:

$$
\begin{aligned}
b &= cq_1 + r_1, & 0 < r_1 < c,\\
c &= r_1 q_2 + r_2, & 0 < r_2 < r_1,\\
r_1 &= r_2 q_3 + r_3, & 0 < r_3 < r_2,\\
&\cdots & \cdots\\
r_{j-2} &= r_{j-1} q_j + r_j, & 0 < r_j < r_{j-1},\\
r_{j-1} &= r_j q_{j+1}.
\end{aligned}
$$

The gcd (b,c) of b and c is r_j, the last nonzero remainder in the division process. Values of x_0 and y_0 in $(b,c) = bx_0 + cy_0$ can be obtained by writing each r_i as a linear combination of b and c.

$$(b,c) = (c,b) = (c, b - cq_1) = (c, r_1)$$

$$(c, r_1) = (r_1, c) = (r_1, c - r_1 q_2) = (r_1, r_2)$$

$$(r_1, r_2) = (r_2, r_1) = (r_2, r_1 - r_2 q_3) = (r_2, r_3)$$

$$\cdots$$

$$(r_{j-2}, r_{j-1}) = (r_{j-1}, r_{j-2}) = (r_{j-1}, r_{j-2} - r_{j-1} q_j) = (r_{j-1}, r_j)$$

$$(r_{j-1}, r_j) = (r_j q_{j+1}, r_j) = (r_j, r_j q_{j+1}) = (r_j, r_j q_{j+1} - r_j q_{j+1})$$

$$= (r_j, 0) = \underline{r_j}.$$

The remainders $r_1, r_2, r_3, \ldots, r_j$ are strictly decreasing. Therefore the algorithm will terminate in a finite number of steps.

<u>Extended Euclidean algorithm</u>: (finding x_0 and y_0 such that $(b,c) = bx_0 + cy_0$.

let $Z_{b,c} = \{ bx + cy \mid x, y \in \mathbb{Z} \}$. We have:

$r_1 = b - cq_1 \Rightarrow r_1 \in Z_{b,c}$

$r_2 = c - r_1 q_2 \Rightarrow r_2 \in Z_{b,c}$

$r_3 = r_1 - r_2 q_3 \Rightarrow r_3 \in Z_{b,c}$

$\vdots$

$r_j = r_{j-2} - r_{j-1} q_j \Rightarrow r_j \in Z_{b,c}$

$\Rightarrow r_j = (b,c) = bx_0 + cy_0$

<u>Example</u>: Let $b = 963$ and $c = 657$.

$963 = 657(1) + 306$

$657 = 306(2) + 45$

$306 = 45(6) + 36$

$45 = 36(1) + \boxed{9}$

$36 = 9(4)$

$(963, 657) = 9$

$306 = 963(1) - 657(1)$

$45 = 657(1) - 306(2) = 657(1) - 963(2) + 657(2)$

$= 963(-2) + 657(3)$

$36 = 306 - 45(6) = 963(1) - 657(1) + 963(12)$

$- 657(18) = 963(13) - 657(19)$

$9 = 45 - 36(1) = 963(-2) + 657(3) - 963(13) + 657(19)$

$= \underline{963(-15) + 657(22)}$

Complexity of Euclid's algorithm:

```
Euclid (a, b)
  if b == 0
      return a
  else return Euclid (b, a - b⌊a/b⌋)
```

Assume that $a > b \geq 0$. If $b > a \geq 0$, then Euclid(a, b) immediately makes the recursive call Euclid(b, a).

If $a > b \geq 1$ and the call Euclid(a, b) performs $k \geq 1$ recursive calls, then $a \geq F_{k+2}$ and $b \geq F_{k+1}$.

Proof by induction: Basis: $k = 1$. $b \geq 1 = F_2$,

$a > b \Rightarrow a \geq 2 = F_3$.

Since $b > a - b\lfloor \frac{a}{b} \rfloor$, in each recursive call the first argument is strictly larger than the second; the assumption that $a > b$ therefore holds for each recursive call.

Induction step: Assume that the statement is true for upto $k-1$ recursive calls. Assume that Euclid(a, b) makes k recursive calls. $k > 0 \Rightarrow b > 0$ and Euclid(a, b) calls Euclid$(b, a - b\lfloor \frac{a}{b} \rfloor)$ recursively, which in turn makes $k-1$ recursive calls.

$\Rightarrow$ $b \geq F_{k+1}$ and $a - b\lfloor \frac{a}{b} \rfloor \geq F_k$.

$b + (a - b\lfloor \frac{a}{b} \rfloor) = a + b(1 - \lfloor \frac{a}{b} \rfloor) \leq a$

Since $a > b > 0 \Rightarrow \lfloor \frac{a}{b} \rfloor \geq 1$.

$a \geq b + (a - b\lfloor \frac{a}{b} \rfloor) \geq F_{k+1} + F_k = F_{k+2}$

For any integer $k \geq 1$, if $a > b \geq 1$ and $b < F_{k+1}$, then the call Euclid(a, b) makes fewer than k recursive calls.

Euclid (F_{k+1}, F_k) makes exactly $k-1$ recursive calls when $k \geq 2$. Proof by induction. Basis: $k=2$. Euclid (F_3, F_2) makes exactly one recursive call to Euclid $(1,0)$. For the induction step, assume that Euclid (F_k, F_{k-1}) makes exactly $k-2$ recursive calls. For $k>2$, we have $F_k > F_{k-1} > 0$ and $F_{k+1} = F_k + F_{k-1}$,

$$\Rightarrow F_{k+1} - F_k \left\lfloor \frac{F_{k+1}}{F_k} \right\rfloor = F_{k-1} \Rightarrow (F_{k+1}, F_k)$$

$$= \left(F_k, F_{k+1} - F_k \left\lfloor \frac{F_{k+1}}{F_k} \right\rfloor\right) = (F_k, F_{k-1})$$

$\Rightarrow$ Euclid (F_{k+1}, F_k) recurses one time more than the call Euclid (F_k, F_{k-1}) which is exactly $(k-1)$ times. $F_k \approx \left(\frac{\sqrt{5}+1}{2}\right)^k \Rightarrow$ Number of recursive calls is $O(\log b)$. Therefore, if we call Euclid on two β-bit numbers, then it performs $O(\beta)$ arithmetic operations and $O(\beta^3)$ bit operations (assuming that multiplication and division of β-bit numbers take $O(\beta^2)$ bit operations).