Extended Euclidean Algorithm (xgcd).

Fact: the gcd of $a, b \in \mathbb{Z}$ is the smallest positive non-zero element of this set:

$$S = \{xa + yb \mid x, y \in \mathbb{Z}\}.$$

Sketch: say $d = x^* a + y^* b$ is the smallest element of S. Note then that d divides all elements of S:

$$(xa + yb) = qd + r, \quad r \lneq d$$

But $\Rightarrow r = 0$, as d was minimal:

$$xa + yb - qd = r$$

$$= a(x - qx^*) + b(y - qy^*)$$

(so, $r \in S$.)

$$\therefore \; d = \gcd(a, b). \checkmark$$

Goal for us: find x, y s.t.

$$d = \gcd(a, b) = xa + yb.$$

Application: "modular inverses":

given $a \in \{1, \ldots p-1\}$ for some large prime p,

find $\in \{1, 2, \ldots p-1\}$ s.t. $a \cdot x \,\%\, p = 1$ $\quad x \equiv a^{-1}$.

To find x, we could just apply our goal to a, p.

$$xa + yp = 1 \Rightarrow xa = 1 - yp$$

$$xa \% p = (1 - yp) \% p = 1$$

Details

inputs: a, b

outputs: d, x, y.

```
                                         outputs
                                           ↙    ↘
int xgcd(int a, int b, int& x , int & y)
{
      if (b==0) {
           x = 1;
           y = 0;
           return a;  // d = a = 1·a + 0·b
      }

    // Imagine xgcd works for all smaller
    // inputs (smaller values of b)
  // How could the answer to xgcd(b, a%b, _, _)
  // help us?
  int x', y';
  int d = xgcd(b, a%b, x', y');
  // d = x'·b + y'·r     (r = a%b)
  // how are x', y' useful???
  // (Remember: we want x,y ∈ ℤ s.t. d = xa + yb.
```

Set $a = qb + r$ $\quad (q = a/b \quad r = a \% b)$

So, $\underbrace{a - qb} = r$

$$d = x'b + y'r$$
$$= x'b + y'(a - qb)$$
$$= \underbrace{y'}_{x}a + \underbrace{(x' - y'q)}_{y}b$$

```
x = y';
y = x' - (a/b) · y'
return d;
}
```