Algorithm

Effective procedure

Unambiguous

Finite

Terminate

Program

Translation into instructions to implement procedure

Step by step

Programming Language

Basic operations

How to combine them

gcd(m,n)

Generate list of common factors of m,n*

Report largest value common to both lists

*Check each k from 1.. min(m,n) as factor of both

```
function gcd(m,n)
    let limit = min(m,n)
    let commonfactors = []
    for each k in 1..limit
        if k|m and k|n
            append k to commonfactors
    report rightmost value in commonfactors
```

gcd_euclid(m,n)

Assume $m > n$

$m, n \geq 1$

If $n | m$

report n

else

report gcd(a,b)

where

$a = \max(n, m-n)$

$b = \min(n, m-n)$

Finite? Termination?

gcd(63, 14)

↳ gcd(49, 14)

↳ gcd(21, 14)

↳ gcd(14, 7)

report 7

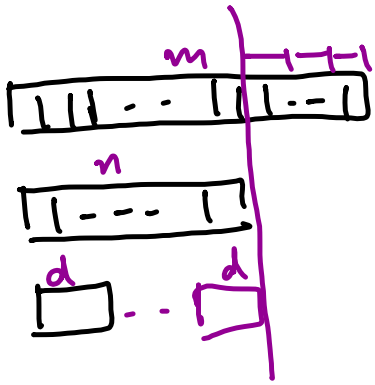
```
function gcd(m,n)
    if n|m
        report n
    else
        let a = max(m-n,n)
        let b = min(m-n,n)
        report gcd(a,b)
```

```
gcd(101,2)
  gcd(99,2)
      ⋮
gcd(3,2)
  gcd(2,1)
    report 1
```

If n does not divide m

$$m = an + r$$

$$d|m \quad d|n \Rightarrow d|r$$



More efficient gcd

gcd(m,n)

← Can show this terminates in time log(max(m,n))

if n|m then report n

remainder

else report gcd(n, m mod n)

gcd(101,2) → gcd(1,2)

gcd(2,101) → gcd(101,2)

In Python

```
def gcd(m,n):
    limit = min(m,n)
    commonfactors = []
    for k in range(1, limit+1):
        if m%k == 0 and n%k == 0:
            commonfactors.append(k)
    return (commonfactors[-1])
```

```
def gcd(m,n):
    limit = min(m,n)
    ~~commonfactors = []~~
    for k in range(1, limit+1):
        if m%k == 0 and n%k == 0:
            mrcf = k
            ~~commonfactors.append(k)~~
    return (~~commonfactors[-1]~~ mrcf)
```

mrcf
most recent
common factor

Python → Python 2 2.7
→ Python 3 3.4 3.2

Python 3.2

Text editors

emacs, vim

gedit, kate, nano, ...

Python program

```
def fun1(..,..):
    ≡

def fun2(..):
    ≡

⋮
def funk(..):
    ≡

≡
```

```
def gcd(m,n):
    ≡

read values k,l
print gcd(k,l)
```

Control flow

How to give a sequence of instructions

Loops for
while

Condition if
else

Function definition

Data

Names

Types of values

Allowed operations

In Python

Dynamic type — Type of name is determined by latest value assigned to it

Strict type — Type at present determines legal operation