Greatest Common Measure: the Last 2500 Years

Alexander Stepanov

This lecture was originally prepared as the 1999 Arthur Schoffstall Lecture in Computer Science and Computer Engineering at the Rensselaer Polytechnic Institute

Pythagoras (572BC - 497BC)

Plimpton 322

“He attached supreme importance to the study of arithmetic, which he advanced and took out of the region of commercial utility.”

Aristoxenus

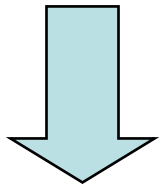
He maintained that “the principles of mathematics are principles of all existing things.”

Aristotle

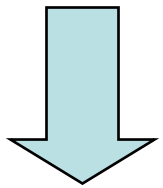
He discovered “the theory of irrationals and the construction of cosmic bodies.”

Proclus

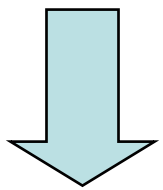
ASTRONOMY



GEOMETRY



NUMBER THEORY



MUSIC

To reduce the world to numbers, one needs the absolute common measure, the smallest physically possible segment, the quantum of space.

IT DOES NOT EXIST!

However small a measure we pick there are segments that cannot be measured by it.

$$gcm(a, a) = a$$

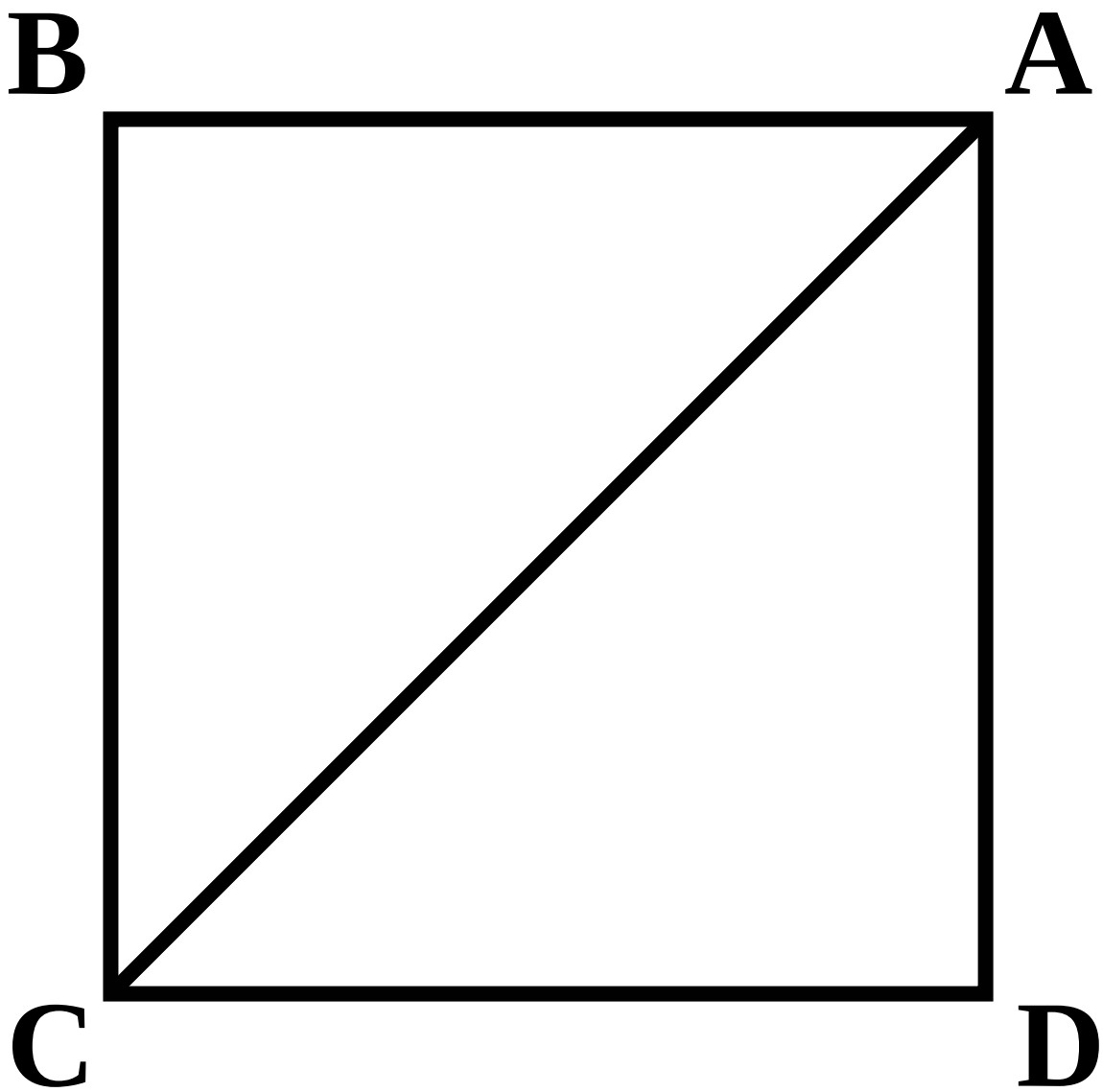
$$gcm(a, b) = gcm(a, a + b)$$

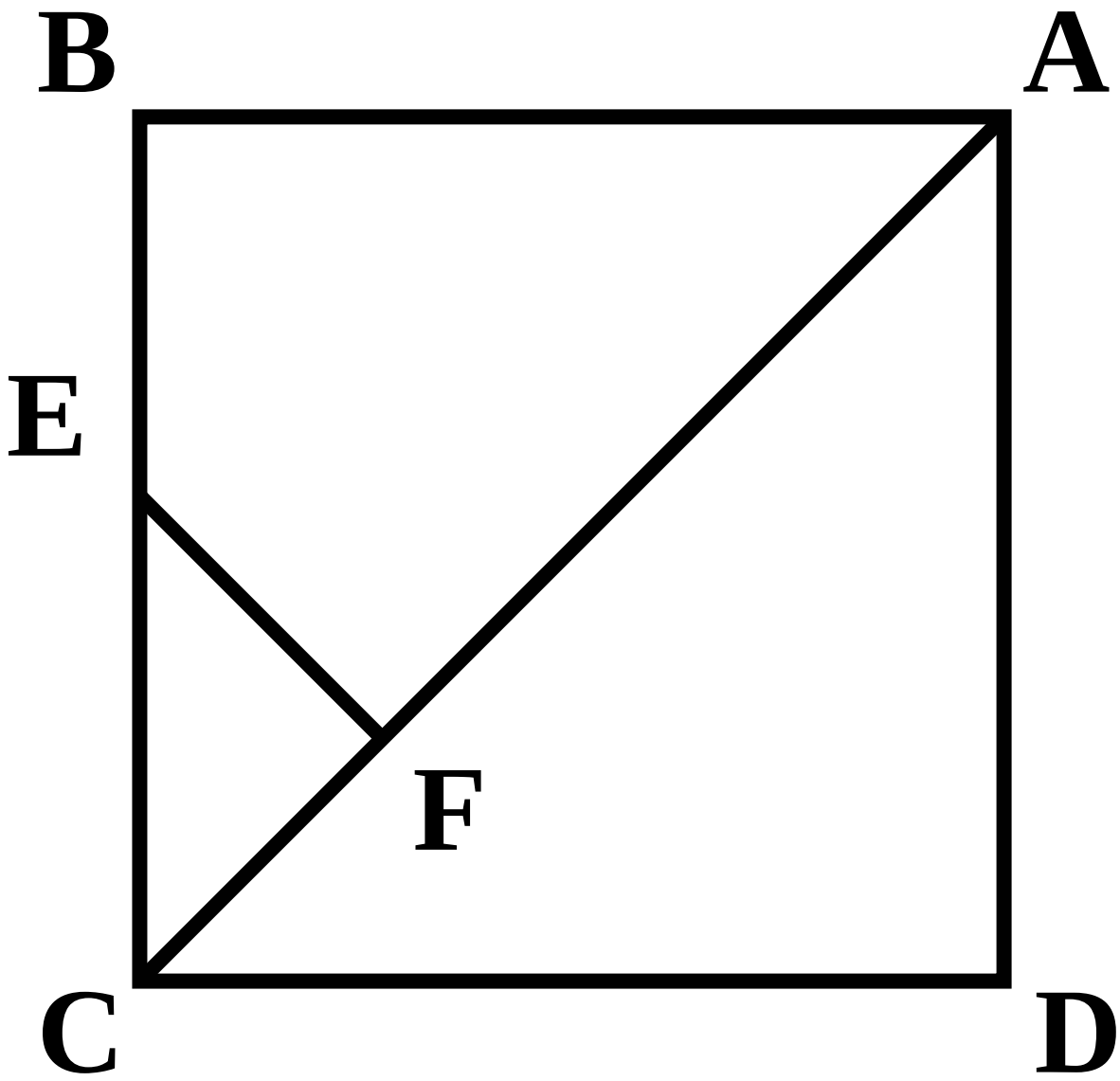
$$a > b \Rightarrow gcm(a, b) = gcm(a - b, b)$$

$$gcm(a, b) = gcm(b, a)$$

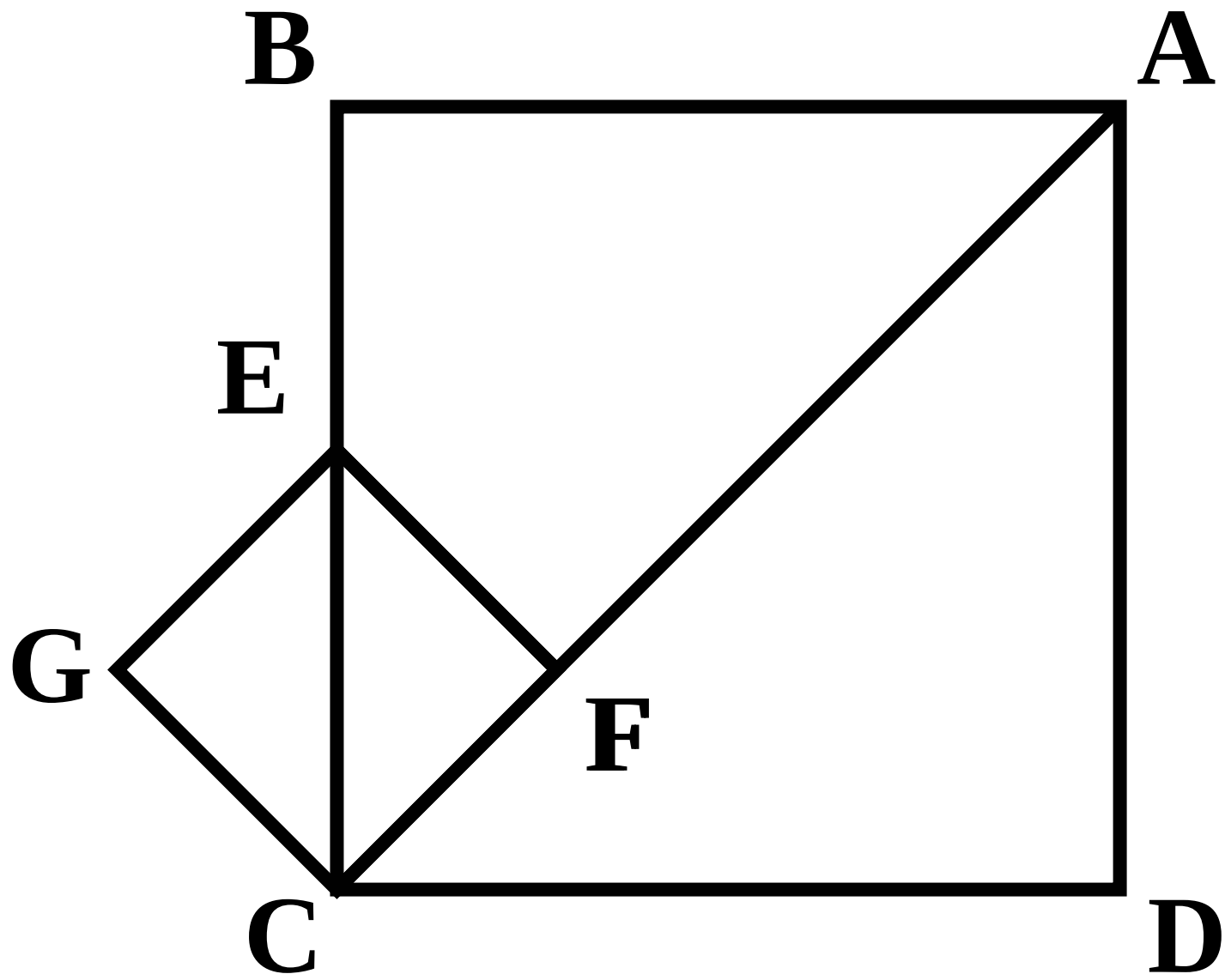
```
line_segment gcm(line_segment a,
                 line_segment b) {
    if (a == b)   return a;
    if (a > b)    return gcm(a-b, b);
/*  if (a < b) */ return gcm(a, b-a);
}
```

Let us assume that there is a measure that can measure both the side and the diagonal of some square. Let us take the smallest such square for the given measure.

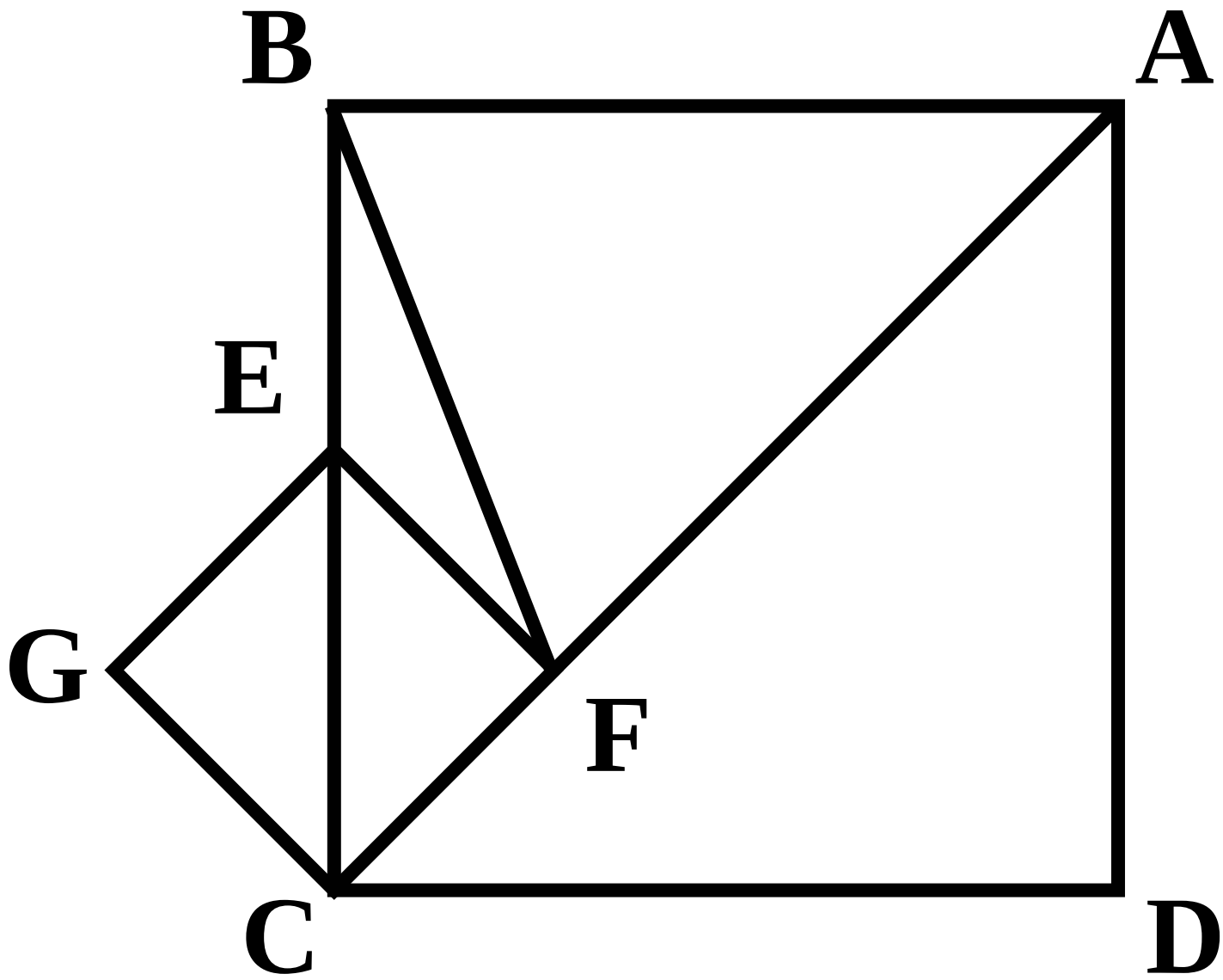

B
A
C
D



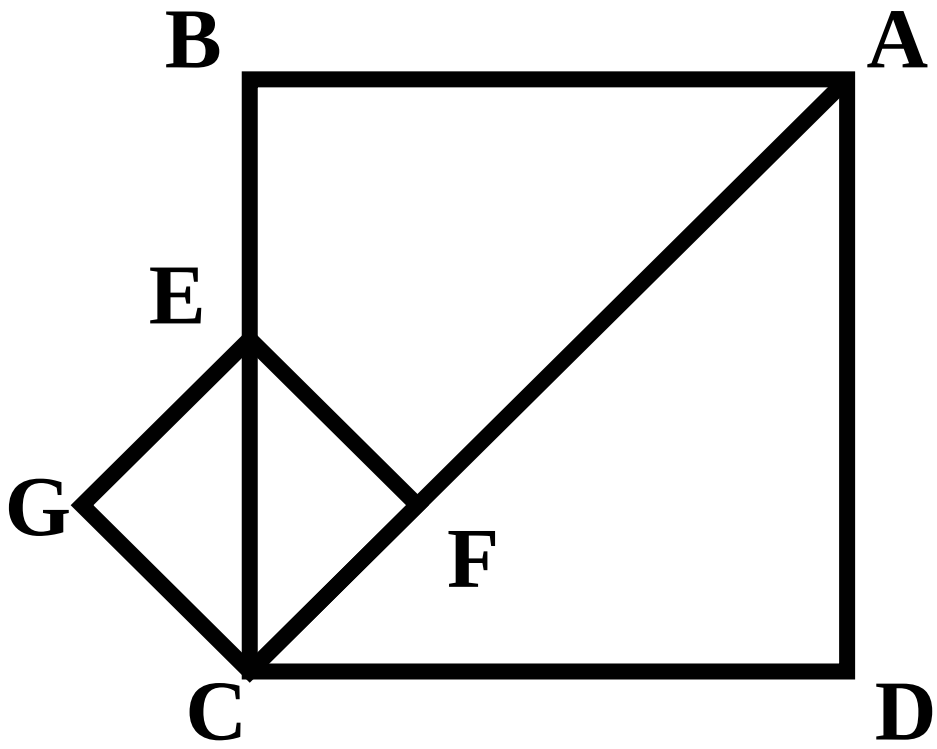
AB = AF **AC ⊥ EF**



$AC \perp CG$ $EG \perp EF$

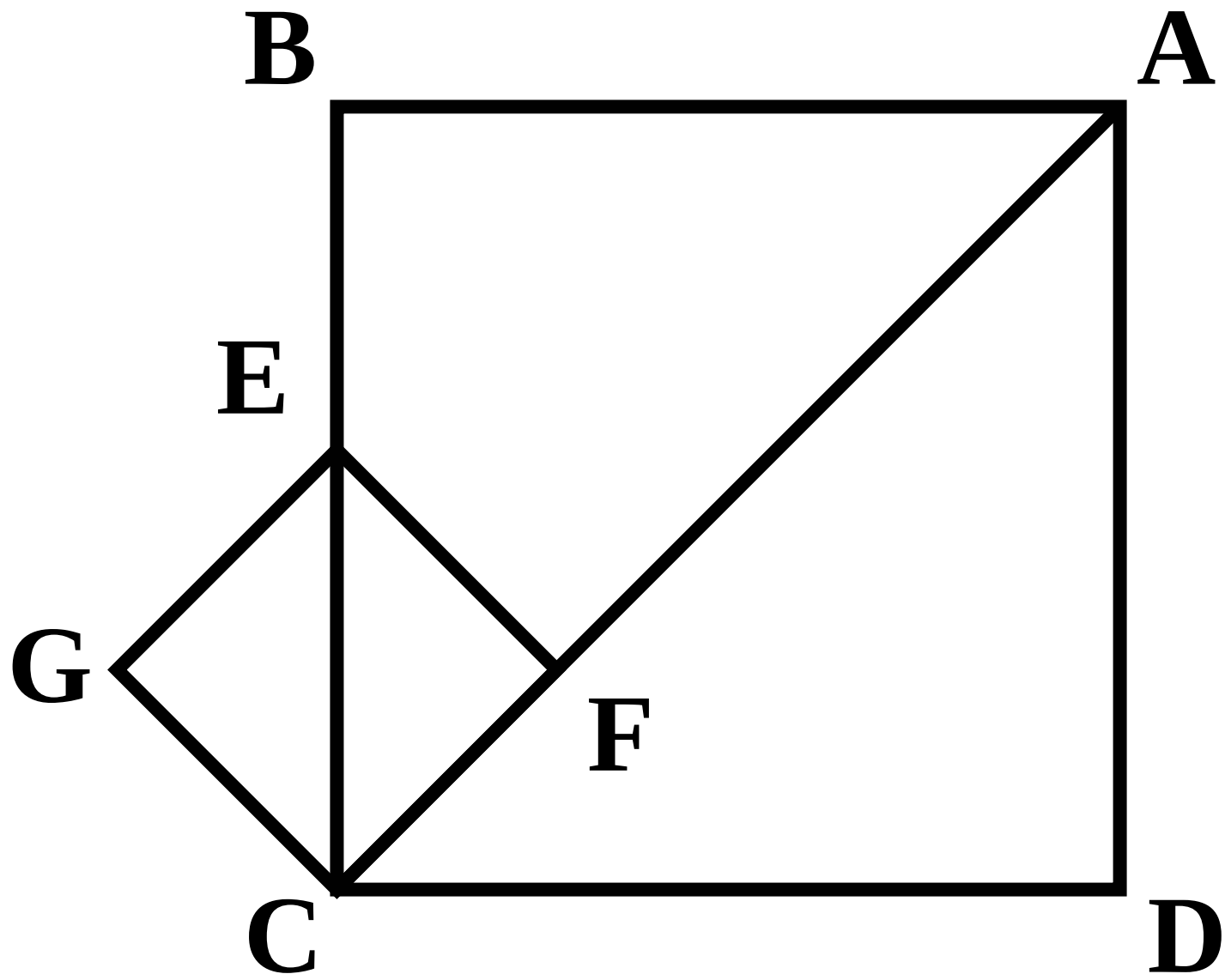


$$CF = EF = EB$$



gcm(AC,AB) = gcm(AB,CF)

gcm(CE,CF) = gcm(AC,AB)



$$EC > EB \rightarrow EB < AB/2$$

We constructed a smaller square that could be measured by this measure. Contradiction!

The side and the diagonal of the square produced by the proof result from two iterations of the algorithm:

```
d, s ->  s, d - s ->  2s - d, d - s
```

And the original ratio is repeated:

```
d/s = (2s - d)/(d -s)
```

Plato (427BC - 347BC)

ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤΩ

LET NO ONE WHO DOES NOT KNOW GEOMETRY ENTER

They came [to Plato's lecture on the Good] in the conviction that they would get some one or other of the things that the world calls good: riches, or health, or strength. But when they found that Plato's reasonings were of mathematics their disenchantment was complete.

Aristoxenus

Plato's Algorithm

```
void sqrt_of_2(int count) {
  int side = 1;
  int diagonal = 1;

  for(int n = 0; n < count; ++n) {
    int tmp = side + diagonal;
    diagonal = tmp + side;
    side = tmp;
  }

  display(diagonal, side, count);
}
```

Rational diameter of:

```
1 is 1              (1)
2 is 3              (1.5)
5 is 7              (1.4)
12 is 17            (1.41667)
29 is 41            (1.41379)
70 is 99            (1.41429)
169 is 239          (1.4142)
408 is 577          (1.41422)
985 is 1393         (1.41421)
2378 is 3363        (1.41421)
```

Euclid (325BC-365BC)

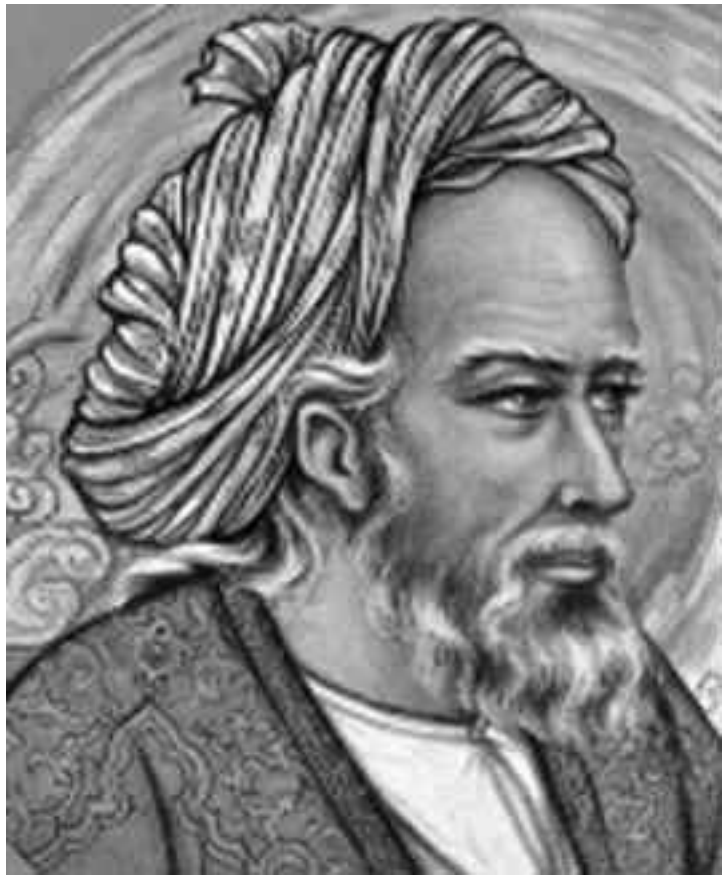
Euclid guaranteed termination by changing the input types:

```
unsigned int gcd(unsigned int a,
                 unsigned int b) {
   assert(a > 0 && b > 0);
   // should wait for Arabs
   // and Leonardo Pisano
   if (a == b)   return a;
   if (a > b)    return gcd(a-b, b);
/* if (b > a) */ return gcd(a, b-a);
}
```

Euclid guaranteed termination by changing the input types:

```
unsigned int gcd(unsigned int a,
                 unsigned int b) {
   assert(a > 0 && b > 0);
   // should wait for Arabs
   // and Leonardo Pisano
   if (a == b)   return a;
   if (a > b)    return gcd(a-b, b);
/* if (b > a) */ return gcd(a, b-a);
}
```

Why `a - b`, not `a % b`?



Omar Khayyam (1048-1123)

```
Sqrt(2) = 1 + 1/(2 + 1/(1 + 1/(2 + ...
```

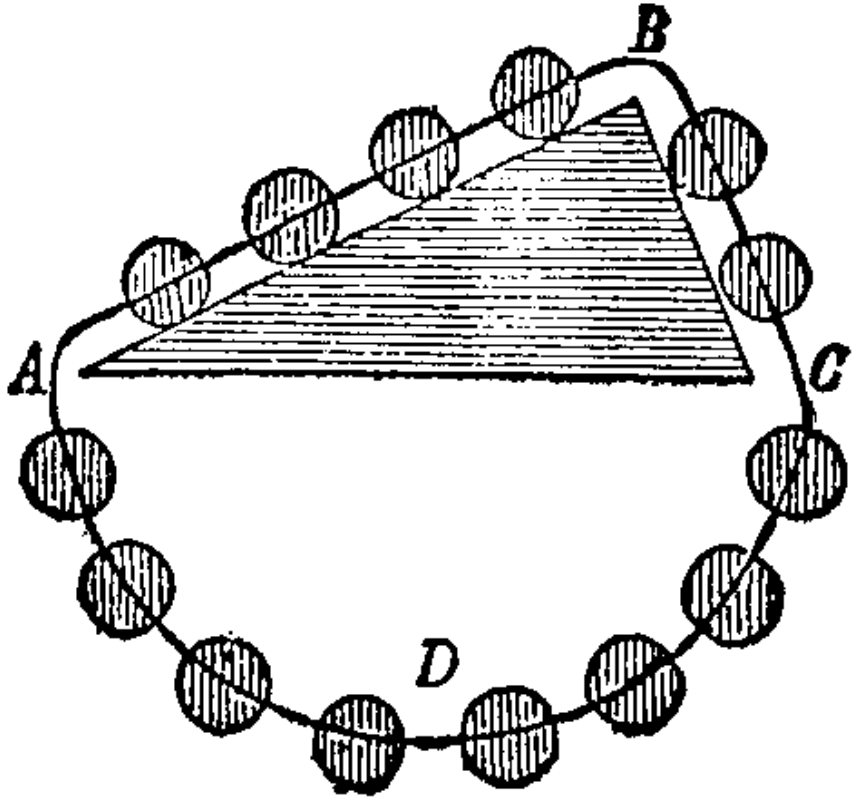
Continued fractions generated by quotients represent a ratio of any two segments.

Leonardo Pisano (1170-1250)

It took over 1500 years to move to:

```
unsigned int gcd(unsigned int m,
                 unsigned int n) {
   while (n != 0) {
     unsigned int t = m % n;
     m = n;
     n = t;
   }
   return m;
}
```

Simon Stevin (1548 - 1620)


B
A
C
D

Simon Stevin:

```
int gcd(int m, int n) {
    while (n != 0) {
      int t = m % n;
      m = n;
      n = t;
    }
    return m;
}
```

Simon Stevin:

```
Polynomial<real>
gcd(Polynomial<real> m,
    Polynomial<real> n) {
   while (n != 0) {
     polynomial<real> t = m % n;
     m = n;
     n = t;
   }
   return m;
}
```

Carl Friedrich Gauss
(1777 - 1855)

Given many numbers *A*, *B*, *C*, etc the *greatest common divisor* is found as follow. Let all the numbers be resolved into their prime factors, and from these extract the ones which are common to *A*, *B*, *C*, etc…

Gauss, *Disqisitiones Arithmeticae*, art. 18

Carl Gauss:

```
complex<int>
gcd(complex<int> m,
    complex<int> n) {
   while (n != 0) {
     complex<int> t = m % n;
     m = n;
     n = t;
   }
   return m;
}
```

Lejeune Dirichlet
(1805 - 1857)

“It is now clear that the whole structure of number theory rests on a single foundation, namely the algorithm for finding the greatest common divisor of two numbers.”

Lejeune Dirichlet, *Vorlesungen über Zahlentheorie*

Richard Dedekind
(1831 - 1916)

Emmy Noether (1882 -1935)

It was she who taught us to think in terms of simple and general algebraic concepts – homomorphic mappings, groups and rings with operators, ideals…

P.S. Alexandrov

For Emmy Noether, relationships among numbers, functions, and operations became transparent, amenable to generalisation, and productive only after they have been dissociated from any particular objects and have been reduced to general conceptual relationships…

B.L. van der Waerden

Bartel Leendert van der Waerden
(1903 - 1996)

Dedekind, Noether, van der Waerden:

```
template <class
          EuclideanRingElement>
EuclideanRingElement
gcd(EuclideanRingElement m,
    EuclideanRingElement n) {
   while (n != 0) {
     EuclideanRingElement t = m % n;
     m = n;
     n = t;
   }
   return m;
}
```

Donald Knuth

(1938 -)

Knuth's objection: gcd(1, -1) = -1

```
template <class
          EuclideanRingElement>
EuclideanRingElement
gcd(EuclideanRingElement m,
    EuclideanRingElement n) {
   while (n != 0) {
     EuclideanRingElement t = m % n;
     m = n;
     n = t;
   }
   if (m < 0) m = -m;
   return m;
}
```

Depends on the definition!

Greatest common divisor is a common divisor that is divisible by any other common divisor.

What is Euclidian Domain?

- What are operations and their requirements?
- What are intended models?
- What are related algorithms?

Extended Euclid

```
template <class EuclideanDomain>
triple<EuclideanDomain, EuclideanDomain, EuclideanDomain>
extended_euclid(EuclideanDomain u, EuclideanDomain v) {
   EuclideanDomain u0 = 1;
   EuclideanDomain v0 = 0;
   EuclideanDomain u1 = u;
   EuclideanDomain v1 = v;
   while (v1 != 0) {
        EuclideanDomain q = u1/v1;
        u0 -= v0 * q;
        swap(u0, v0);
        u1 -= v1 * q;
        swap(u1, v1);
   }
   return make_triple(u0, (u1 - u * u0) / v, u1);
}
```

Josef Stein (1961):

$$\gcd(n, 0) = \gcd(0, n) = n$$
$$\gcd(n, n) = n$$
$$\gcd(2n, 2m) = 2\gcd(n, m)$$
$$\gcd(2n, 2m + 1) = \gcd(n, 2m + 1)$$
$$\gcd(2n + 1, 2m) = \gcd(2n + 1, m)$$
$$\gcd(2n + 1, 2(n + k) + 1) = \gcd(2(n + k) + 1, 2n + 1) = \gcd(2n + 1, k)$$

```
template <class BinaryInteger>
BinaryInteger gcd(BinaryInteger m,
                  BinaryInteger n) {

  make_non_negative(m);
  make_non_negative(n);

  if (is_zero(m)) return n;
  if (is_zero(n)) return m;

  int d = 0;

  while (is_even(m) && is_even(n)) {
    half_non_negative(m);
    half_non_negative(n);
    ++d;
  }
```

```
  while (is_even(m)) half_non_negative(m);

  while (is_even(n)) half_non_negative(n);

  while (true)
    if (m < n) {
      n = n - m;
      do {
        half_non_negative(n);
      } while (is_even(n));
    } else if (n < m) {
      m = m - n;
      do {
        half_non_negative(m);
      } while (is_even(m));
    } else
      return left_shift(m, d);
}
```

What is Stein domain?

David Fowler, *The Mathematics Of Plato's Academy*, Oxford, 1999

John Stillwell, *Mathematics and Its History*, Springer-Verlag, 1989

John Stillwell, *Elements of Number Theory*, Springer-Verlag, 2002

Euclid, *Elements*, translated by Sir Thomas L. Heath, Dover , 1956 (3 volumes)

Robin Hartshorne, *Geometry: Euclid and Beyond*, Springer-Verlag, 2000

Laurence E. Siegler, *Fibonacci's Liber Abaci*, Springer-Verlag, 2002

Nicolas Bourbaki, *Elements of the History of Mathematics*, Springer-Verlag, 1999

Carl Friedrich Gauss, *Disquisitiones Arithmaticae*, Yale, 1965

Peter Gustav Lejeune Dirichlet, *Lectures on Number Theory*, AMS, 1999

Richard Dedekind, *Theory of Algebraic Integers*, Cambridge, 1996

B. L. van der Waerden, *Algebra*, Springer-Verlag, 1994

Donald Knuth, *Art of Computer Programming, vol. 2, Seminumerical Algorithms*, Addison-Wesley, 1998

Josef Stein, *Computational problems associated with Racah algebra*, J. Comput. Phys., 1, 397-405

Andre Weilert, *(1+ i)-ary GCD Computation in Z[i] as an Analogue of the Binary GCD Algorithm*, J. Symbolic Computation (2000) 30, 605-617