[an error occurred while processing this directive]

[an error occurred while processing this directive]

Euclid's Algorithm

Euclid's Algorithm appears as the solution to the Proposition VII.2 in the **Ipiqirxw**:

> **Kmzir x{s ryqfivw rsx tvmqi xs sri ersxliv0 xs ārh xlimv kviexiwx gsqqsr qiewyvi2**

What Euclid called "common measure" is termed nowadays a **gsqqsr jegxsv sv e gsqqsr hmzmwsv ,lxxtw>33{{{2gyx1xli1orsx2svk3evmxlqixmg3JegxsvwErhQypxmtpiw2wlxqp-**. Euclid VII.2 then offers an **epksvmxlq ,lxxtw>33{{{2gyx1xli1orsx2svk3[lexMw3[lexMwEpksvmxlq2wlxqp-** for finding the **kviexiwx gsqqsr hmzmwsv ,lxxtw>33{{{2gyx1xli1orsx2svk3fpyi3glmiwi2wlxqp'kgh-** (gcd) of two integers. Not surprisingly, the algorithm bears Euclid's name.

The algorithm is based on the following two observations:

1. If b|a then **kgh ,lxxtw>33{{{2gyx1xli1orsx2svk3fpyi3glmiwi2wlxqp-**(a, b) = b.

   This is indeed so because no number (b, in particular) may have a divisor greater than the number itself (I am talking here of non-negative integers.)

2. If a = bt + r, for integers t and r, then gcd(a, b) = gcd(b, r).

   Indeed, every common divisor of a and b also divides r. Thus gcd(a, b) divides r. But, of course, gcd(a, b)|b. Therefore, gcd(a, b) is a common divisor of b and r and hence gcd(a, b) ≤ gcd(b, r). The reverse is also true because every divisor of b and r also divides a.

Example

Let a = 2322, b = 654.

2322 = 654·3 + 360	gcd(2322, 654) = gcd(654, 360)
654 = 360·1 + 294	gcd(654, 360) = gcd(360, 294)
360 = 294·1 + 66	gcd(360, 294) = gcd(294, 66)
294 = 66·4 + 30	gcd(294, 66) = gcd(66, 30)
66 = 30·2 + 6	gcd(66, 30) = gcd(30, 6)
30 = 6·5	gcd(30, 6) = 6

Therefore, gcd(2322,654) = 6.

For any pair a and b, the algorithm is bound to terminate since every new step generates a similar problem (that of finding gcd) for a pair of smaller integers. Let Eulen(a, b) denote the length of the Euclidean algorithm for a pair a, b. Eulen(2322, 654) = 6, Eulen(30, 6) = 1. I'll use this notation in the proof of the following very important consequence of the algorithm:

Corollary

> **Jsv iziv} temv sj {lspi ryqfivw e erh f xlivi evi x{s mrxikivw w erh x wygl xlex ew/ fxA kgh,e0f-2**

Example

2322×20 + 654×(-71) = 6.

Proof

Let a > b. The proof is by induction on Eulen(a, b). If Eulen(a, b) = 1, i.e., if b|a, then a = bu for an integer u. Hence, a + (1 - u)b = b = gcd(a, b). We can take s = 1 and t = 1 - u.

Assume the Corollary has been established for all pairs of numbers for which Eulen is less than n. Let Eulen(a, b) = n. Apply one step of the algorithm: a = bu + r. Eulen(b, r) = n - 1. By the inductive assumption, there exist x and y such that bx + ry = gcd(b,r) = gcd(a,b). Express r as r = a - bu. Hence, ry = ay - buy; bx + (ay - buy) = gcd(a, b). Finally, b(x - uy) + ay = gcd(a, b) and we can take s = x - uy and t = y.

There is also a **wmq t pi t vssj ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3t mki sr l spi 3l ygpih2wl xq p** that employs the **Tmki sr l spi Tvmgmtpi ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3hsc}sycor s{ 3t mki sr 2wl xq p**.

Remark

Note that any linear combination as + bt is divisible by any common factor of a and b. In particular, any common factor of a and b also divides gcd(a, b). In a "reverse" application, any linear combination as + bt is divisible by gcd(a, b). From here it follows that gcd(a, b) is the least positive integer representable in the form as + bt. All the rest are multiples of gcd(a, b). The generalization of the Corollary to what is known as **Tvmgmtepmhi ephsq em** is known as **FÐ~syx+wmhi r xmx}** or **FÐ~syx+wPi q q e** after the French mathematician Éttiene Bézout (1730-1783), so it often happens that the result stated in the Corollary is also often referred to as **FÐ~syx+wmhi r xmx}** or **FÐ~syx+wPi q q e**.

For **gst vmq i ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3hsc}sycor s{ 3ji { c{ svhw2wl xq p gst vmq i -** numbers we get existence of s and t such that as + bt = 1. This Corollary is a powerful tool. It appeared in the **7 Kpeww ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3{ exi v62wl xq p** and **Lsyv Kpeww ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3l kcwspyxmsr 2wl xq p** problems. For example, let's prove the Euclid's Proposition VII.30

> **MJx{ s ryq f i w0q ypxmpmh f} sri er sxl i vq eoi wsq i ryq f i v0er h er} t vmq i ryq f i v q i ewyvi wxl i t vshygx0xl i r mxepws q i ewyvi wsri sj xl i svmkmep ryq f i w2**

Let a prime p divide the product ab. Assume p∤a. Then gcd(a, p) = 1. By Corollary, ax + py = 1 for some x and y. Multiply by b: abx + pby = b. Now, p|ab and p|pb. Hence, p|b.

Actually, this proves a generalization of the Proposition VII.30 I used several times on these pages:

> Let m|ab and gcd(a, m) = 1. Then m|b.

Proposition VII.30 immediately implies the Fundamental Theorem of Arithmetic although Euclid has never stated it explicitly. The first time it was formulated in 1801 by Gauss in his **Hmwuymwmxsr i wevmxl q i xmgei** .

Fundamental Theorem of Arithmetic

> **Er} mxi ki vR ger f i vi t vi wi r xi h ewe t vshygxsj t vmq i w2Wygl e vi t vi wi r xexmsr mw yr mu yi yt xs xl i svhi vsj t vmq i jegxsw2**

Since, by definition, a number is **gsq t swmxi** if it has factors other than 1 and itself, and these factors are bound to be smaller than the number, we can keep extracting the factors until only prime factors remain. This shows existence of the representation: N = pqr ..., where all p, q, r,... are prime. To prove uniqueness, assume there are two representations: N = pqr ... = uvw.... We see that p divides uvw... By Corollary, it divides one of the factors u, v, w, ... Cancel them out. We can go on chipping away on the factors left and right until no factors remain.

> **Vi t vi wi r xexmsr sj e ryq f i v ewxl i t vshygxsj t vmq i wmwgepmh t vmq i ryq f i v hi gsq t swmxmsr svt vmq i jegxsvm~exmsr 2Xl i Jyr heq i r xepXl i svi q sj Evmxl q i xmg ewwi vxw xl exi egl mxi ki vl ewe yr mu yi t vmq i ryq f i vhi gsq t swmxmsr 2**

Note: Euclid's Algorithm is not the only way to determine the greatest common factor of two integers. If you can find the prime factorizations of the two numbers you can easily determine their gcd as the **mxi wwi gxmsr sj xl i q ypxmwi xw ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3Gywmgypyq 3Evmxl q i xmg3KGHF}FXvi i 2wl xq p** formed by their prime factors. **Jegxsv Xvi i w ,l xxt w>33{ { { 2gyx1xl i 1or sx2svk3Gywmgypyq 3Evmxl q i xmg3FXvi i Xi wxmk2wl xq p** offer a convenient bookkeeping for finding prime factorizations of integers.

References

1. H. Davenport, **Xl i Lmkl i v Evmxl q i xmg0Levt i v* Fvsxl i w0R]**
   ,l xxt w>33{ { { 2eq e~sr 2gsq 3i |i g3sf mhsw3MWFRA49658666;63gxowsjx{ evi mngE3-
2. **V2Kvel eq 0H2Or yxl 0S2Texewl r mo0**
   ,l xxt w>33{ { { 2eq e~sr 2gsq 3i |i g3sf mhsw3MWFRA49658666;63gxowsjx{ evi mngE3-Gsr gvi xi Qexl i q exmgw
   ,l xxt w>33{ { { 2eq e~sr 2gsq 3i |i g3sf mhsw3MWFRA464599<4693gxowsjx{ evi mngE3-, 2nd edition, Addison-Wesley, 1994.
3. Oystein Ore, **Ryq f i v Xl i sv} er h Mxw Lmwxsv}**
   ,l xxt w>33{ { { 2eq e~sr 2gsq 3i |i g3sf mhsw3MWFRA48<::9:64=3gxowsjx{ evi mngE3-, Dover Publications, 1976

4. S. K. Stein, Qexliqexmgw>Xli Qer1Qehi Yrmzivwi ,lxxt>33{{{2eqe~sr2gsq3i|ig3sfmhsw3MWFRA48<:8489453gxowsjx{evimgE3-, 3rd edition, Dover, 2000.

[an error occurred while processing this directive]

[an error occurred while processing this directive]

ÀGsrxegxÀ,lxxt>33{{{2gyx1xli1orsx2svk3QempRsxmjmgexmsr Teki2wlxqpÀJvsrx tekiÀ,lxxt>33{{{2gyx1xli1orsx2svk3jvsrx2wlxqpÀGsrxirxwÀ,lxxt>33{{{2gyx1xli1orsx2svk3gsrxirx2wlxqpÀEpkifveÀ,lxxt>33{{{2gyx1xli1orsx2svk3epkifve2wlxqp

[an error occurred while processing this directive]

[an error occurred while processing this directive]