A Fibonacci számok

Definíció: A Fibonacci számsorozat

Fibonacci számsorozatnak nevezzük azt az $F_0, F_1, F_2, \ldots$ számsorozatot, amelyet az alábbi formulapár határoz meg:

$$\left.\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \end{aligned}\right\} \textit{(kezdőfeltétel)} \tag{1}$$

$$F_{n+2} = F_{n+1} + F_n, \quad n = 0,1,2,\ldots \textit{ (rekurziós feltétel)} \tag{2}$$

Az (1), (2) formulapár által előállított számok sorozata így kezdődik:

Számok	***0***	***1***	***1***	***2***	***3***	***5***	***8***	***13***	***21***	***34***	***55***	***89***	***144***	***233***	***377***	**…**
Jelölés	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	…

A probléma a fenti definícióval az, hogy az *n* indexű elemet nem tudjuk a megelőzőek nélkül kiszámítani a rekurziós formula alapján. Mennyi például F_{100}-nak az értéke? Ennek a problémának a megoldását adja a Binet formula.

1. Tétel: A Binet formula

A Fibonacci számsorozat elemei felírhatók az index függvényében az alábbi úgynevezett *Binet formula* révén:

$$F_n = \frac{1}{\sqrt{5}}\left(\Phi^n - \overline{\Phi}^n\right), \quad n = 0,1,2,\ldots$$
$$\text{ahol } \Phi = \frac{1+\sqrt{5}}{2} \approx 1.618\ldots, \ \overline{\Phi} = \frac{1-\sqrt{5}}{2} \approx -0.618\ldots \tag{3}$$

Bizonyítás

Keresnünk kell olyan számsorozatot, amely az (1), (2) feltételeknek megfelel. Könnyebb lesz a dolgunk, ha egyelőre az (1) feltételtől eltekintünk. A trükk: keressünk olyan $\{a_n\}$ sorozatokat, amelyek tudják a (2) tulajdonságot és alakjuk $a_n=z^n$ valamilyen *z* számra. A megoldások közül zárjuk ki a *z=0* esetet, mint érdektelent, hiszen ez csupa zérust ad és az nekünk biztosan nem jó. A (2) tulajdonságot felírva a_n-re

$$z^{n+2}=z^{n+1}+z^n, \ n=0,1,2,... \tag{4}$$

adódik, ahonnan z^n-nel leosztva a

$$z^2=z+1 \tag{5}$$

összefüggésre jutunk. Ennek a másodfokú egyenletnek két valós megoldása van: $z_1 = \Phi$ és $z_2 = \overline{\Phi}$. (Ellenőrizzük!) Tehát az $a_n = z_1^n$ megoldás. Könnyen meggyőződhetünk behelyettesítéssel (2)-be, hogy akkor az $a_n = C_1 z_1^n$ is megoldás, ahol C_1 tetszőleges konstans. Ugyanígy mivel $a_n = z_2^n$ megoldás, akkor $a_n = C_2 z_2^n$ is az. Itt C_2 szintén tetszőleges konstans. Azt kaptuk, hogy a megoldások konstansszorosai is megoldások. Vegyük észre továbbá, hogy az

$$a_n = C_1 z_1^n + C_2 z_2^n \tag{6}$$

is megoldás. Összegezve: a z_1^n és a z_2^n úgynevezett alapmegoldások (bázismegoldások) *lineáris kombinációi* is megoldást adnak. Helyettesítsük be (6)-ot (1)-be az *n=0* és *n=1* esetre. Ezzel csempésszük vissza a kezdetben elhanyagolt (1) feltételt. Az alábbi kétismeretlenes, két egyenletből álló lineáris egyenletrendszert kapjuk, melyben az ismeretlenek C_1 és C_2 .

$$\begin{aligned} C_1 &+ C_2 &= 0 \\ C_1\Phi &+ C_2\overline{\Phi} &= 1 \end{aligned} \tag{7}$$

Innen C_1-et és C_2-t kifejezve kapjuk, hogy

$$C_1 = \frac{1}{\sqrt{5}}, \quad C_2 = -\frac{1}{\sqrt{5}} \tag{8}$$

■

2. Tétel: A Fibonacci számok előállítása kerekítéssel

Az F_n Fibonacci szám képezhető az alábbi módon a Binet formula felhasználásával:

$$F_n = Round\left(\frac{1}{\sqrt{5}}\Phi^n\right), \quad n = 0,1,2,\ldots \tag{9}$$

Bizonyítás

Belátjuk, hogy $\frac{1}{\sqrt{5}}\Phi^n$ és F_n között kevesebb, mint $\frac{1}{2}$ az eltérés. Ez azt jelenti, hogy a kerekítendő $\frac{1}{\sqrt{5}}\Phi^n$ szám köré fel tudunk rajzolni egy szimmetrikus intervallumot, amelynek a szélessége kevesebb, mint egy. Ekkor azonban csak egyetlen egész szám lehet ebben az intervallumban, ami így szükségszerűen éppen a Fibonacci szám lesz.

$$\begin{aligned} \left|F_n - \frac{1}{\sqrt{5}}\Phi^n\right| &= \left|\left(\frac{1}{\sqrt{5}}\Phi^n - \frac{1}{\sqrt{5}}\overline{\Phi}^n\right) - \frac{1}{\sqrt{5}}\Phi^n\right| = \left|-\frac{1}{\sqrt{5}}\overline{\Phi}^n\right| \le \\ &\le \frac{1}{\sqrt{5}}\cdot\left|\overline{\Phi}^n\right| \le \frac{1}{\sqrt{5}}\cdot\left|\overline{\Phi}\right|^n \le \frac{1}{\sqrt{5}} < \frac{1}{2} \end{aligned} \tag{10}$$

Ugyanis $\left|\overline{\Phi}\right| < 1$ és $\sqrt{5} > 2$. Tehát

$$\frac{1}{\sqrt{5}}\Phi^n - \frac{1}{2} < F_n < \frac{1}{\sqrt{5}}\Phi^n + \frac{1}{2}, \tag{11}$$

amiből látszik, hogy F_n egybeesik az egyetlen egésszel, amely a megadott intervallumban van.

■

A Fibonacci számok sorozata exponenciálisan növekszik. Ez következik a $F_n = Round\left(\frac{1}{\sqrt{5}}\Phi^n\right), \quad n = 0,1,2,\ldots$ előállításból.

$$F_n \approx 0{,}447213595 \cdot 1{,}618033989^n. \tag{12}$$

Írhatjuk, hogy $F_n = \Theta\left(\Phi^n\right)$.

2. Számelméleti algoritmusok

2.1 Alapfogalmak

Definíció: Az oszthatóság

Azt mondjuk, hogy a *d* egész szám *osztja* az *a* egész számot, ha az osztásnak zérus a maradéka, azaz, ha létezik olyan *k* egész szám, hogy $a=k\cdot d$. Jelölésben: $d \mid a$. A *d* számot az *a* *osztó*jának nevezzük. Az *a* szám a *d* *többszöröse*.

Definíció: Prímszám

Prímszámnak nevezzük azt az *1*-nél nagyobb egész számot, amelynek csak az *1* és saját maga az osztója.

1. Tétel: A maradékos osztás tétele

Ha *a* egész szám, *n* pedig pozitív egész szám, akkor egyértelműen létezik olyan *q* és *r* egész szám, hogy

$$a = q \cdot n + r \text{, ahol } 0 \le r < n. \tag{1}$$

A *q* szám neve *hányados*, *r* neve *maradék*. A hányados és a maradék felírható:

$$q = \lfloor a / n \rfloor,\ r = a - q \cdot n = a \bmod n. \tag{2}$$

A bizonyítást az olvasóra bízzuk.

Definíció: Közös osztó

Azt mondjuk, hogy a *d* egész szám az *a* és *b* egészek közös osztója, ha *d* mindkét számot osztja. $d \mid a$ és $d \mid b$.

Definíció: Lineáris kombináció

Az *s* egész számot az *a* és *b* egészek (egész) *lineáris kombináció*jának nevezzük, ha létezik olyan *x* és *y* egész szám, hogy $s = x \cdot a + y \cdot b$. Az *x* és *y* számokat a lineáris kombináció *együtthatói*nak nevezzük. Az *a* és *b* számok összes lineáris kombinációjának halmazát $L(a,b)$-vel jelöljük.

1. Példa: Legyen két szám *a=36, b=60*. Határozzuk meg az $s = x \cdot a + y \cdot b$ értékeket az $x = -3,\ldots,3$, $y = -3,\ldots,3$ együtthatókra!

s=xa+yb tábla		*y*						
		-3	*-2*	*-1*	*0*	*1*	*2*	*3*
x	*-3*	-288	-228	-168	-108	-48	12	72
	-2	-252	-192	-132	-72	-12	48	108
	-1	-216	-156	-96	-36	24	84	144
	0	-180	-120	-60	0	60	120	180
	1	-144	-84	-24	36	96	156	216
	2	-108	-48	12	72	132	192	252
	3	-72	-12	48	108	168	228	288

2. Tétel: A közös osztó tulajdonságai

Legyen a d egész az a és b egészek közös osztója. Akkor fennállnak az alábbi állítások:

1. $|d| \le |a|$ vagy $a = 0$
2. Ha $d|a$ és $a|d$, akkor $d = \pm a$
3. A közös osztó osztója az a és b szám minden lineáris kombinációjának is. $\forall s \in L(a,b)$-re $d|s$.

A bizonyítást az olvasóra bízzuk.

2.2 A legnagyobb közös osztó

Definíció: Legnagyobb közös osztó

Az alább megadott d^* egész számot az a és b egész számok legnagyobb közös osztójának nevezzük. Jele lnko(a,b).

$$d^* \overset{def}{=} lnko(a,b) = \begin{cases} 0 & ha \;\; a = 0 \;\; és \;\; b = 0 \\ \max\limits_{\substack{d|a \\ d|b}} d & egyébként \end{cases} \qquad (1)$$

Definíció: Relatív prímek

Az a és b egész számokat *relatív prímek*nek nevezzük, ha $lnko(a,b) = 1$.

1. Tétel: A legnagyobb közös osztó elemi tulajdonságai

Legyen a d^* egész az a és b egészek legnagyobb közös osztója. Akkor fennállnak az alábbi állítások:

1. $1 \le d^* \le \min(|a|,|b|)$. (2)
2. $lnko(a,b) = lnko(b,a) = lnko(-a,b) = lnko(|a|,|b|)$. (3)
3. $lnko(a,0) = |a|$. (4)
4. $lnko(a, k \cdot a) = |a|$, $k \in Z$. (5)
5. Ha d közös osztó és $d^* \ne 0$, akkor $d \le d^*$.
6. A legnagyobb közös osztó minden lineáris kombinációnak osztója, azaz $\forall s \in L(a,b)$-re $d^*|s$.

2. Tétel: A legnagyobb közös osztó reprezentációs tétele

Ha az a és b egész számok nem mindegyike zérus, akkor a legnagyobb közös osztó megegyezik a két szám pozitív lineáris kombinációinak minimumával.

$$d^* = lnko(a,b) = \min_{\substack{s \in L(a,b) \\ s>0}} s = a \cdot x^* + b \cdot y^* = s^* \qquad (6)$$

Bizonyítás

A bizonyítás menete az lesz, hogy megmutatjuk, hogy $s^* \le d^*$ és $d^* \le s^*$, amiből következik az állítás.

$s^* \le d^*$ megmutatása úgy történik, hogy belátjuk, hogy s^* közös osztó, ami nem lehet nagyobb, mint a legnagyobb közös osztó. Azt, hogy s^* közös osztó azáltal látjuk be, hogy az osztási maradéka zérus. Csak az a számra végezzük el, b-re ugyanígy megy a bizonyítás. Osszuk el tehát a-t s^*-gal és számítsuk ki a maradékot! Legyen q a hányados. Az r maradékra igaz, hogy $0 \le r < s^*$. Akkor

$$\begin{aligned} &0 \le r < s^* \\ &0 \le a - q \cdot s^* < s^* \\ &0 \le a - q \cdot \left(a \cdot x^* + b \cdot y^*\right) < s^* \\ &0 \le \left(1 - q \cdot x^*\right) \cdot a + \left(-q \cdot y^*\right) \cdot b < s^* \end{aligned} \tag{7}$$

Az egyenlőtlenség közepén álló maradék az a és a b lineáris kombinációja. A baloldali egyenlőtlenség miatt nem lehet negatív, a jobboldali egyenlőtlenség miatt pedig kisebb, mint a pozitív lineáris kombinációk közül a legkisebb. Emiatt csak zérus lehet. Tehát az s^* osztja az a számot.

$d^* \le s^*$ abból következik, hogy a legnagyobb közös osztó osztja az összes lineáris kombinációt, így s^*-ot is. Ekkor azonban nem lehet nagyobb, mint s^*, hiszen annak osztója.

■

A tétel következményei:

1. A közös osztó osztja a legnagyobb közös osztót, ugyanis a legnagyobb közös osztó az a és b egészek lineáris kombinációja a tétel szerint, amit a közös osztó oszt.
2. Tetszőleges n nemnegatív egészre

$$lnko(n \cdot a, n \cdot b) = n \cdot lnko(a,b). \tag{8}$$

Ugyanis $n = 0$-ra az állítás triviális, $n > 0$-ra pedig
$lnko(n \cdot a, n \cdot b) = n \cdot a \cdot x^* + n \cdot b \cdot y^* = n \cdot \left(a \cdot x^* + b \cdot y^*\right) = n \cdot lnko(a,b)$

(Lássuk be, hogy az utolsó egyenlőségjel valóban igaz, azaz a lineáris kombinációkban mindkét számpár esetén ugyanaz az x^*, y^* megfelelő választás!)

3. Tétel: <u>A lineáris kombinációk halmazának jellemzése</u>
Legyen M a $d^* = lnko(a,b)$ egész többszöröseinek a halmaza. Állítás:

$$L(a,b) \equiv M. \tag{9}$$

Bővebben: az $L(a,b)$ minden eleme d^* egész többszöröse és ha egy s szám d^* egész többszöröse, akkor az az s szám az a és b lineáris kombinációja is.

Bizonyítás

Megmutatjuk, hogy $L \subset M$ és $M \subset L$, amiből következik az állítás.

$L \subset M$ esete: $d^* | a$ és $d^* | b \Rightarrow$ van olyan $k_a, k_b \in Z$, hogy $a = k_a \cdot d^*$, $b = k_b \cdot d^*$.

Ha $s \in L$, akkor $s = a \cdot x + b \cdot y = k_a \cdot d^* \cdot x + k_b \cdot d^* \cdot y = d^* \cdot \underbrace{\left(k_a \cdot x + k_b \cdot y\right)}_{=k} =$

$= k \cdot d^* \in M$, mert $k \in Z$

$M \subset L$ esete: $d^* = lnko(a,b) = a \cdot x^* + b \cdot y^*$

Ha $s \in M$, akkor van olyan $k_s \in Z$, hogy

$s = k_s \cdot d^* = k_s \cdot \left(a \cdot x^* + b \cdot y^*\right) = k_s \cdot x^* \cdot a + k_s \cdot y^* \cdot b \in L$

■

4. Tétel: A legnagyobb közös osztó redukciós tétele

Tetszőleges *a* és *b* két egész szám esetén fennáll, hogy

$$lnko(a,b) = lnko(a-b,b). \tag{10}$$

Bizonyítás

Legyen $d_1^* = lnko(a,b)$ és $d_2^* = lnko(a-b,b)$. Azt fogjuk bizonyítani, hogy $d_1^* | d_2^*$ és $d_2^* | d_1^*$, amiből következik a tétel állítása.

Világos, hogy fennállnak az alábbi tulajdonságok

$$\begin{aligned} &d_1^*\text{-ra:} \quad d_1^* \in L(a,b), \qquad d_1^* | L(a,b) && (10)\\ &d_2^*\text{-ra:} \quad d_2^* \in L(a-b,b), \quad d_2^* | L(a-b,b) && (11) \end{aligned}$$

Megmutatjuk, hogy $d_1^* \in L(a-b,b)$ és $d_2^* \in L(a,b)$ is fennáll. amiből a kölcsönös oszthatóság már következik.

$d_1^* \in L(a-b,b)$ megmutatása:

$$\begin{aligned} d_1^* \in L(a,b) \quad \Rightarrow \quad d_1^* &= x_1^* \cdot a + y_1^* \cdot b = x_1^* \cdot (a-b+b) + y_1^* \cdot b = \\ &= x_1^* \cdot (a-b) + \left(x_1^* + y_1^*\right) \cdot b \in L(a-b,b) \end{aligned}$$

$d_2^* \in L(a,b)$ megmutatása:

$$\begin{aligned} d_2^* \in L(a-b,b) \quad \Rightarrow \quad d_2^* &= x_2^* \cdot (a-b) + y_2^* \cdot b = \\ &= x_2^* \cdot a + \left(y_2^* - x_2^*\right) \cdot b \in L(a,b) \end{aligned}$$

■

2.3. A bináris legnagyobb közös osztó algoritmus

Az eddigiek alapján algoritmus konstruálható a legnagyobb közös osztó meghatározására. Az algoritmus neve: *Bináris lnko algoritmus*.

Az algoritmus a két nemnegatív egész bináris felírásának alakjából indul ki. Az utolsó bit alapján a kiinduló problémát fokozatosan egyszerűbbé redukálja, amíg csak az egyik szám zérussá nem válik. Ekkor a legnagyobb közös osztó a másik redukált szám egy szorzóval korrigált értéke lesz. Munka közben az algoritmus csak egyszerű, hatékony gépi műveleteket - egész kivonás és jobbra eltolás (shift) – használ.

Legyen a két szám *a* és *b* és legyen $a \geq b$. $Lnko(a,b)$ kiszámításának feladata ekkor az alábbiak szerint redukálódik az utolsó bit szerint egyszerűbb feladattá:

Utolsó bit: *a* \ *b*	*0*	*1*
0	$2 \cdot lnko(a/2, b/2)$	$lnko(a/2, b)$
1	$lnko(a, b/2)$	$lnko((a-b)/2, b)$

(Bizonyítsuk be a táblázatbeli egyszerűsítések helyességét!)
A bináris lnko algoritmus pszeudokódja:

2.3.1 algoritmus Bináris legnagyobb közös osztó	
	//
1	Bináris_lnko (a, b, d^*)
2	// Input paraméter: a∈Z, a≥0
3	// b∈Z, b≥0, a≥b
4	// Output paraméter: d^*∈Z, d^*≥0, d^*=lnko(a,b)
5	$c \leftarrow 1$
6	**WHILE** $a \neq 0$ *és* $b \neq 0$ **DO**
7	// *a* és *b* paritásértékei p_a és p_b. *0* = páros, *1* = páratlan
8	$p_a = a \bmod 2$
9	$p_b = b \bmod 2$
10	**CASE**
11	$p_a=0, p_b=0$: $c \leftarrow 2c$
11	$a \leftarrow a/2$
12	$b \leftarrow b/2$
13	$p_a=0, p_b=1$: $a \leftarrow a/2$
14	$p_a=1, p_b=0$: $b \leftarrow b/2$
15	$p_a=1, p_b=1$: $a \leftarrow (a-b)/2$
16	**IF** $a < b$ **THEN** $a \leftrightarrow b$ *csere*
17	**IF** $a = 0$
18	**THEN** $d^* \leftarrow c \cdot b$
19	**ELSE** $d^* \leftarrow c \cdot a$
20	**RETURN** (d^*)

1. Példa: példa az algoritmusra: *Lnko(3604,3332)=$2^2 \cdot 17=68$*

Lépésszám	a		b	Korrekciós szorzó
0	3604		3332	2
1	1802		1666	2
2	901		833	
3	34	↔	833	
4	833		34	
5	833		17	
6	408		17	
7	204		17	
8	102		17	
9	51		17	
10	17		17	
11	0		17	

2.4. Az euklideszi és a kibővített euklideszi algoritmus

1. Tétel: A legnagyobb közös osztó rekurziós tétele

Tetszőleges *a* és *b* két egész szám esetén fennáll, hogy

$$lnko(a,b) = lnko(b, a \bmod b). \qquad (1)$$

Bizonyítás

Az *a* mod *b* az *a*-ból *b* ismételt kivonásaival megkapható és így a *redukciós tétel* értelmében az állításunk igaz.

■

A rekurziós tétel révén készíthető el az euklideszi algoritmus a legnagyobb közös osztó meghatározására. Az algoritmus pszeudokódja:

2.4.1. algoritmus Euklideszi algoritmus	
	//
	// rekurzív változat
1	Euklidesz (a, b, d^*)
2	// Input paraméter : a∈Z, a≥0
3	// b∈Z, b≥0
4	// Output paraméter: d^*∈Z, d^*≥0
5	$d^* \leftarrow$ a
6	**IF** $b \neq 0$
7	**THEN** Euklidesz (b, a mod b, d^*)
8	**RETURN** (d^*)

2.4.2. algoritmus	
Euklideszi algoritmus	
	//
	// iteratív változat
1	Euklidesz (a, b, d^*)
2	// Input paraméter : a∈Z, a≥0
3	// b∈Z, b≥0
4	// Output paraméter: d^*∈Z, d^*≥0
5	**WHILE** $b \neq 0$ **DO**
6	r←a mod b
7	a←b
8	b←r
9	d^*←a
10	**RETURN** (d^*)

1. Példa: *Lnko(3604,3332)=68,* $q = \lfloor a/n \rfloor,\ r = a - q \cdot n = a \bmod n$

Lépésszám	a	b	q	r
0	3604	3332	1	272
1	3332	272	12	68
2	272	68	4	0
3	68	0	-	68

2. Tétel: Lamé tétele

Ha az euklideszi algoritmusban $a > b \geq 0$ és $b < F_{k+1}$ valamely $k > 0$-ra, akkor a rekurziós hívások száma kevesebb, mint *k*.

A tételt nem bizonyítjuk.

Következmény a tételből: Ha $F_k \leq b < F_{k+1}$, akkor a rekurziós hívások száma kevesebb, mint *k*, valamint becslést tudunk adni erre a *k*-ra közvetlenül a *b*-ből. A *k* értékére jól memorizálható becslés az, hogy *k* vehető a *b* tizes számrendszerbeli jegyei ötszörösének. (Valójában megmutatható, hogy itt a kisebb reláció is igaz.) A meggondolás az alábbi:

$$F_k \approx \frac{1}{\sqrt{5}} \Phi^k,\ \Phi \approx 1{,}618,\ \log_{10} \Phi \approx 0{,}2089,\ \frac{1}{\log_{10} \Phi} \approx 4{,}78 \approx 5 \tag{2}$$

$$\log_{10} F_k \approx k \cdot \log_{10} \Phi - \log_{10} \sqrt{5} \tag{3}$$

$$k \approx \frac{1}{\log_{10} \Phi} \cdot \log_{10} F_k + \frac{\log_{10} \sqrt{5}}{\log_{10} \Phi} \approx 5 \cdot \log_{10} F_k \approx 5 \cdot \log_{10} b \tag{4}$$

Bizonyítható, hogy az euklideszi algoritmusnak a legrosszabb bemenő adatai a szomszédos Fibonacci számok. Az euklideszi algoritmus időigénye $O(\log b)$ azon feltételezés mellett, hogy az aritmetikai műveletek konstans ideig tartanak függetlenül a benne szereplő számértékek nagyságától. Ha a számok nagyságát is figyelembe vesszük, akkor az időigény $O(\log a \cdot \log b)$. Az euklideszi algoritmus némi bővítéssel alkalmassá tehető arra, hogy a legnagyobb közös osztó lineáris kombinációként történő előállításában szereplő x^* és y^* együtthatókat is meghatározza. Tekintsük az euklideszi táblát.

Lépésszám	a	b	q	r
0	a_0	b_0	q_0	r_0
1	a_1	b_1	q_1	r_1
…	…	…	…	…
k	a_k	b_k	q_k	r_k
k+1	a_{k+1}	b_{k+1}	q_{k+1}	r_{k+1}
…	…	…	…	…
n	d^*	0	-	d^*

A tábla *k.* sorában $(k = 0,1,\ldots,n)$ a *rekurziós tétel* alapján érvényes a $d^* = x_k^* \cdot a_k + y_k^* \cdot b_k$ összefüggés, ahol továbbá $a_{k+1} = b_k$, $b_{k+1} = r_k$, $q_k = \lfloor a_k / b_k \rfloor$, $r_k = a_k - q_k \cdot b_k$. A *k* és *k+1* indexű sorok között a kapcsolat:

$$
\begin{aligned}
d^* &= x_k^* \cdot a_k + y_k^* \cdot b_k = \\
&= x_k^* \cdot (q_k \cdot b_k + r_k) + y_k^* \cdot b_k = \\
&= (x_k^* \cdot q_k \cdot b_k + x_k^* \cdot r_k) + y_k^* \cdot b_k = \\
&= (x_k^* \cdot q_k + y_k^*) \cdot b_k + x_k^* \cdot r_k = \\
&= x_{k+1}^* \cdot a_{k+1} + y_{k+1}^* \cdot b_{k+1}
\end{aligned}
\tag{5}
$$

Kaptunk egy összefüggést az x_k^*, y_k^* és az x_{k+1}^*, y_{k+1}^* együtthatók között az egymást követő sorokra, ha fentről lefelé haladunk a táblában.

$$
\begin{aligned}
x_{k+1}^* &= x_k^* \cdot q_k + y_k^* \\
y_{k+1}^* &= x_k^*
\end{aligned}
\tag{6}
$$

Haladjunk most lentről fölfelé! Akkor (6)-ból x_k^* és y_k^* kifejezve:

$$
\begin{aligned}
x_k^* &= y_{k+1}^* \\
y_k^* &= x_{k+1}^* - q_k \cdot y_{k+1}^*
\end{aligned}
\tag{7}
$$

Az utolsó sor esetén viszont $d^* = 1 \cdot d^* + 0 \cdot 0$, azaz $x_n^* = 1$ és $y_n^* = 0$. Az utolsó sorból indulva így visszafelé sorról-sorra haladva az x_k^* és y_k^* értékek kiszámíthatók. Végül az x_0^* és y_0^* is kiadódik. Ez a módosítás vezet az euklideszi algoritmus kibővítésére, melynek pszeudokódját alább közöljük.

	2.4.3. algoritmus **Kibővített euklideszi algoritmus**
	// rekurzív változat
1	Kibővített_Euklidesz (a, b, d^*, x^*, y^*)
2	// Input paraméterek : a,b∈Z, a,b≥0
3	// Output paraméterek: d^*, x^*, y^*∈Z, d^*≥0
4	**IF** $b = 0$
5	**THEN** $d^* \leftarrow a$
6	$x^* \leftarrow 1$
7	$y^* \leftarrow 0$
8	**ELSE** Kibővített_Euklidesz (b, a mod b, d^*, x^*, y^*)
9	$\begin{pmatrix} x^* \\ y^* \end{pmatrix} \leftarrow \begin{pmatrix} y^* \\ x^* - \lfloor a/b \rfloor \cdot y^* \end{pmatrix}$
10	**RETURN** (d^*, x^*, y^*)

2.4.3. algoritmus **Kibővített euklideszi algoritmus**	
	// iteratív változat
1	Kibővített_Euklidesz (a, b, d^*, x^*, y^*)
2	// Input paraméterek : a,b∈Z, a,b≥0
3	// Output paraméterek: d^*, x^*, y^*∈Z, d^*≥0
4	$x_0 \leftarrow 1, x_1 \leftarrow 0, y_0 \leftarrow 0, y_1 \leftarrow 1, s \leftarrow 1$
5	**WHILE** $b \neq 0$
6	$r \leftarrow a \bmod b, q \leftarrow a \operatorname{div} b$
7	$a \leftarrow b, b \leftarrow r$
8	$x \leftarrow x_1, y \leftarrow y_1$
9	$x_1 \leftarrow q \cdot x_1 + x_0, y_1 \leftarrow q \cdot y_1 + y_0$
10	$x_0 \leftarrow x, y_0 \leftarrow y$
11	$s \leftarrow -s$
12	$x \leftarrow s \cdot x_0, y \leftarrow -y_0$
13	$(d^*, x^*, y^*) \leftarrow (a, x, y)$
14	**RETURN** (d^*, x^*, y^*)

2. Példa: példa a kibővített euklideszi algoritmusra:

Lépésszám	A	b	q	r	d^*	x^*	y^*
0	3604	3332	1	272	68	-12	1-1·(-12)=13
1	3332	272	12	68	68	1	0-12·1=-12
2	272	68	4	0	68	0	1-4·0=1
3	68	0	-	68	68	1	0

Az algoritmus eredményeképpen $x_0^* = -12$ és $y_0^* = 13$ adódott.

Ellenőrzésképpen 68=(-12)·3604+13·3332=-43248+43316, ami valóban megfelel az elvárásoknak.

FELADATOK

1. Bizonyítsuk be a maradékos osztás tételét!
2. Adjuk meg az összes olyan b pozitív egész számot, amely 100-nál nem nagyobb és teljesül rá, hogy $7 = b \bmod 11$!
3. Adjunk olyan pozitív számpárokat (két különböző szám), amelyek 100-nál nem nagyobbak és az ilyen tulajdonságú párok között a legtöbb közös osztóval rendelkeznek!
4. Soroljuk fel a 30 és a 105 számok összes pozitív, 100-nál nem nagyobb lineáris kombinációját. Adjunk megfelelő együtthatókat is mindegyik lineáris kombinációhoz!
5. Bizonyítsuk be a közös osztó tulajdonságainak tételét!
6. Soroljuk fel az összes pozitív, 100-nál kisebb számot, amely relatv prím a 30-as számmal!
8. Bizonyítsuk be a legnagyobb közös osztó elemi tulajdonságainak tételét!
9. Határozzuk meg a 28560 és a 38640 legnagyobb közös osztóját a bináris legnagyobb közös osztó algoritmusával! Végezzük el a számításokat bináris számrendszerben is!

10. Kiadjuk az Euklidesz(38640 ; 28560, d^*) pszeudokód utasítást, melyben a 2.4.1. algoritmust használjuk. Szemléltessük a parancs végrehajtásának a menetét, a verem alakulását!
11. Adjunk felső becslést a rekurzív hívások számára a 10. feladathoz a Lamé tétel következménye alapján!