Diszkrét matematika
4. előadás

MÁRTON Gyöngyvér
mgyongyi@ms.sapientia.ro

Sapientia Egyetem,
Matematika-Informatika Tanszék
Marosvásárhely, Románia

2023, őszi félév

Miről volt szó az elmúlt előadáson?

- a `tuple`, a `list`, az `str` adattípusok, szeletelések
- adatbevitel, `print` - a kimenet formázása, kiíratás szövegállományba
- hibakezelés, függvények paraméterátadása
- természetes számok, egész számok
- gyorshatványozás
- algoritmusok futási ideje

Miről lesz szó?

- legnagyobb közös osztó,
- a kiterjesztett euklideszi algoritmus,
- diofantoszi egyenletek,
- a `strip`, `split`, `zip`, `enumerate` függvények,

Egész számok

- halmazjelölés: $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$,
- tulajdonságok:
  - $\mathbb{N} \subset \mathbb{Z}$, és a két halmaz számossága ugyanaz,
  - két halmaz számossága, akkor egyezik meg, ha a két halmaz között létezik egy bijekció, pl: $f(x) = \begin{cases} 2x & \text{ha } x \geq 0, \\ -2x - 1 & \text{ha } x < 0, \end{cases}$
  - kommutatívitás, asszociativitás, disztributivitás,
  - az egész számok halmaza zárt az összeadásra, kivonásra, szorzásra nézve, de ez nem igaz az osztásra,
  - az összeadásra nézve minden elemnek van inverz eleme,
  - rendezettség: $a \leq b$, ha $b - a \in \mathbb{N}$.

Oszthatósági problémák

Legyen $a, b \in \mathbb{Z}$, ahol $b \neq 0$. a osztja b-t, ha létezik olyan c egész szám, amelyre fennáll: $b = a \cdot c$, jelölése: $a \mid b$. Azt az esetet, amikor a nem osztja b-t $a \nmid b$-vel jelöljük.

1. tétel (A maradékos osztás tétele)

Legyen a egy egész szám, c pedig egy pozitív egész szám. Ekkor egyértelműen létezik két olyan q és r egész szám, amelyekre igaz $a = c \cdot q + r$, $0 \leq r < c$.

- Az a és b egész számok legnagyobb közös osztója az a legnagyobb pozitív egész szám, amely osztja az a-t és b-t is. Jelölése: $lnko(a, b)$, (a, b), vagy $\gcd(a, b)$.
- Az a és b egész számok legkisebb közös többszöröse az a legkisebb pozitív egész szám amely osztható a-val és b-vel is. Jelölése: $lkkt(a, b)$, vagy $[a, b]$.
- Az a és b egész számok szorzata egyenlő az a és b legnagyobb közös osztójának és legkisebb közös többszörösének a szorzatával:

$$(a, b) \cdot [a, b] = a \cdot b.$$

Oszthatósági problémák

- A legnagyobb közös osztó meghatározására több algoritmus is ismert, az egyik az euklideszi algoritmus. Python-ban a `math` modulban található `gcd` függvény használható két szám legnagyobb közös osztójának a meghatározására:

```
>>> from math import gcd
>>> gcd(60, 45)
    15
>>> gcd(1789, 100)
    1
>>> gcd(-63, 45)
    9
```

- Azt mondjuk, hogy a és b relatív prímek, ha a legnagyobb közös osztójuk 1. A fenti példában 1789 és 100 relatív prímek, míg 60 és 45, illetve -63 és 45 nem.

2. tétel

Legyenek a, b, q, r egész számok, $a = b \cdot q + r$ és legyen $(a, b) = d$, azaz a és b legnagyobb közös osztója d. Ekkor igaz az, hogy: $d = (a, b) = (b, a - b \cdot q)$.

Megjegyzés:

`a - b * q` egyenlő `a % b`-vel.

Az euklideszi algoritmus

Legyenek a, b pozitív egész számok, ahol $a \geq b$ és legyen $r_0 = a$, $r_1 = b$:

$$
\begin{array}{rcll}
r_0 & = & r_1 \cdot q_1 + r_2 & 0 \leq r_2 < r_1, \\
r_1 & = & r_2 \cdot q_2 + r_3 & 0 \leq r_3 < r_2, \\
 & . & & \\
 & . & & \\
 & . & & \\
r_{n-2} & = & r_n - 1 \cdot q_{n-1} + r_n & 0 \leq r_n < r_{n-1}, \\
r_{n-1} & = & r_n \cdot q_n. &
\end{array}
$$

$$
\begin{array}{rcll}
(32, 76) & = & (76, 32 - 0 \cdot 76) & = (76, 32) \\
(76, 32) & = & (32, 76 - 2 \cdot 32) & = (32, 12) \\
(32, 12) & = & (12, 32 - 2 \cdot 12) & = (12, 8) \\
(12, 8) & = & (8, 12 - 1 \cdot 8) & = (8, 4) \\
(8, 4) & = & (4, 8 - 2 \cdot 4) & = (4, 0) = 4
\end{array}
$$

A legnagyobb közös osztó egyenlő lesz a fenti számítási sorozat során meghatározott utolsó nem nullás maradékkal:

$$(a, b) = (r_0, r_1) = (r_1, r_2) = \ldots = (r_{n-2}, r_{n-1}) = (r_{n-1}, r_n) = (r_n, r_0) = r_n.$$

Az algoritmus futási idejét az osztások száma határozza meg, ezzel kapcsolatban megadható a következő tétel:

3. tétel (Lamé tétele)

Az euklideszi algoritmus során végzett osztások száma nem lesz nagyobb, mint ötször a kisebbik szám számjegyeinek száma

Az euklideszi algoritmus

1. feladat

Írjunk egy Python függvényt, amely az euklideszi algoritmus szerint meghatározza két egész szám legnagyobb közös osztóját, de ne használjunk beépített függvényeket.

```
def lnkoF_(a, b):
    while b != 0:
        r = a % b
        a = b
        b = r
    return a

>>> lnkoF(-32, -76)
    4
>>> lnkoF(32.7, 76)
    'hibás bemenet'
```

```
def lnkoF(a, b):
    if not isinstance(a, int) or not isinstance(b, int):
        return 'hibás bemenet'
    a = abs(a)
    b = abs(b)
    while True:
        r = a % b
        if r == 0: break
        a = b
        b = r
    return b
```

A kiterjesztett euklideszi algoritmus

- a d legnagyobb közös osztó valamely x, y egész számokkal mindig felírható az a és b lineáris kombinációjaként:
$$d = x \cdot a + y \cdot b,$$
- az euklideszi algoritmus meghatározza a d legnagyobb közös osztót, a kiterjesztett euklideszi algoritmus pedig az x, y egész számokat is,
- a kiterjesztett euklideszi algoritmus futási ideje logaritmikus éppen ezért fontos szerepet tölt be a számítógépes adatbiztonság területén

Legyenek a, b pozitív egész számok, ekkor felírható a következő összefüggés:

$$(a, b) = x_n \cdot a + y_n \cdot b,$$

ahol x_n és y_n a következő számítási sorozat n-ik tagjai, ahol $j \in \{2, 3, \ldots, n\}$,

$$\begin{aligned} x_0 &= 1, \ y_0 = 0 \\ x_1 &= 0, \ y_1 = 1 \\ x_j &= x_{j-2} - q_{j-1} \cdot x_{j-1} \\ y_j &= y_{j-2} - q_{j-1} \cdot y_{j-1}, \end{aligned}$$

Megállapítható, hogy minden $j = \{2, 3, \ldots, n\}$-re igaz a következő is:

$$r_j = x_j \cdot a + y_j \cdot b.$$

Az iteráció során az r_j osztási maradékokat jelöli, a q_j értékek pedig az osztási egész részeket.

A kiterjesztett euklideszi algoritmus, példa

Határozzuk meg 76 és 32 legnagyobb közös osztóját, majd azokat az x és y egész számokat, melyekre fennáll a következő összefüggés: $76 \cdot x + 32 \cdot y = d$, ahol $d = (76,\ 32)$.

a	b	q	r	x_0	x_1	y_0	y_1	
				1	0	0	1	
76	32	2	12	0	1	1	-2	$1 \cdot 76 + (-2) \cdot 32 = 12$
32	12	2	8	1	-2	-2	5	$(-2) \cdot 76 + 5 \cdot 32 = 8$
12	8	1	4	-2	3	5	-7	$3 \cdot 76 + (-7) \cdot 32 = 4$
8	4	2	0					

Tehát a megoldás: $d = 4,\ x = 3,\ y = -7$, és fennáll a következő összefüggés: $3 \cdot 76 + (-7) \cdot 32 = 4$.

A kiterjesztett euklideszi algoritmus, példa

Határozzuk meg 15 és 56 legnagyobb közös osztóját, majd azokat az x és y egész számokat, melyekre fennáll a következő összefüggés: $15 \cdot x + 56 \cdot y = d$, ahol $d = (15,\ 56)$.

a	b	q	r	x_0	x_1	y_0	y_1
				1	0	0	1
15	56	0	15	0	1	1	0
56	15	3	11	1	-3	0	1
15	11	1	4	-3	4	1	-1
11	4	2	3	4	-11	-1	3
4	3	1	1	-11	15	3	-4
3	1	3	0				

Tehát a megoldás: $d = 1,\ x = 15,\ y = -4$, és fennáll a következő összefüggés: $15 \cdot 15 + 56 \cdot (-4) = 1$.

A kiterjesztett euklideszi algoritmus

2. feladat

Írjunk egy Python függvényt, amely a kiterjesztett euklideszi algoritmusával határozza meg az a és b legnagyobb közös osztóját, a d-t és azokat az x, y egész számokat amelyekre fennáll: $d = x \cdot a + y \cdot b$.

```
def extEuclid(a, b):
    x0, x1, y0, y1 = 1, 0, 0, 1
    while True:
        q = a // b
        r = a - b * q
        if r == 0:
            return b, x1, y1
        x = x0 - q * x1
        y = y0 - q * y1
        x0, x1, y0, y1 = x1, x, y1, y
        a, b = b, r

>>> extEuclid(15, 56)
    (1, 15, -4)
```

A kiterjesztett euklideszi algoritmus

Kiegészítjük az előző függvényt, hogy kiírassuk a részértékeket:

```
def extEuclid_(a, b):
    x0, x1, y0, y1 = 1, 0, 0, 1
    print("%4s%4s%4s%4s%4s%4s%4s%4s" % ("a","b","q","r","x0","x1","y0","y1"))
    print("%4s%4s%4s%4s%4i%4i%4i%4i" % ("","","","",x0,x1,y0,y1))
    while True:
        q = a // b
        r = a - b * q
        if r == 0:
            print("%4i%4i%4i%4i" % (a,b,q,r))
            return b, x1, y1
        x = x0 - q * x1
        y = y0 - q * y1
        x0, x1, y0, y1 = x1, x, y1, y
        print("%4i%4i%4i%4i%4i%4i%4i%4i" % (a,b,q,r,x0,x1,y0,y1))
        a, b = b, r

>>> extEuclid_(15, 56)
    a   b   q   r  x0  x1  y0  y1
                   1   0   0   1
    ...
```

Diofantoszi egyenletek

- egész együtthatós többismeretlenes algebrai egyenletek, amelyeknek megoldásai egész számok (ritkán természetes vagy racionális számok),
- elnevezése Diophantosz (3. század), görög matematikus után

4. tétel

Legyenek a, b egész számok, úgy hogy $lnko(a, b) = d$. Ha $d \not| c$, azaz ha d nem osztja c-t, akkor az $a \cdot x + b \cdot y = c$ egyenletnek nincs megoldása az egész számok körében, Ha $d \mid c$, akkor az egyenletnek végtelen sok megoldása van.

- első lépésben, kiterjesztett euklideszi algoritmussal meghatározzuk a $d, \hat{x}, \hat{y}$-t, úgy hogy teljesüljön: $a \cdot \hat{x} + b \cdot \hat{y} = d$,
- az egyenlet első megoldása: $x_0 = \hat{x} \cdot (c/d), y_0 = \hat{y} \cdot (c/d)$,
- ezek szerint fennáll:

$$\begin{array}{rcl} a \cdot x_0 + b \cdot y_0 & = & c \\ a \cdot x_0 + a \cdot b/d + b \cdot y_0 - a \cdot b/d & = & c \\ a \cdot (x_0 + b/d) + b \cdot (y_0 - (a/d) & = & c \end{array}$$

- az egyenlet többi megoldásait az $n = \ldots, -2, -1, 0, 1, 2, \ldots$ egész számok alapján a következőképpen számoljuk ki:

$$x = x_0 + n \cdot (b/d), y = y_0 - n \cdot (a/d).$$

Diofantoszi egyenletek

- nagyon gyakran egy elsőfokú, kétismeretlenes diofantoszi egyenlet pozitív megoldásait kell meghatározni
- Például: Egy elárusító 1676 ron értékben rendelt almát és körtét. Minden láda alma 36 ronba, és minden láda körte 50 ronba kerül. Hány láda almát és hány láda körtét rendelhetett?
- tulajdonképpen a $36 \cdot x + 50 \cdot y = 1676$ diofantoszi egyenlet pozitív megoldásait kell megkeresnünk.
- egy $a \cdot x + b \cdot y = c$ egyenlet pozitív megoldásai esetében fenn kell álljon:

$$x_0 + n \cdot \frac{b}{d} \geq 0 \Rightarrow n \geq \frac{-x_0}{\frac{b}{d}}$$

$$y_0 - n \cdot \frac{a}{d} \geq 0 \Rightarrow n \leq \frac{y_0}{\frac{a}{d}}$$

- keressük tehát azokat az n természetes számokat, amelyek kielégítik a fenti két egyenlőtlenséget

Diofantoszi egyenletek, példa

A $36 \cdot x + 50 \cdot y = 1676$ diofantoszi egyenlet pozitív megoldásait keressük, ahol a megoldás menete a következő:

- a kiterjesztett euklideszi algoritmussal meghatározzuk: $d = 2, \hat{x} = 7, \hat{y} = -5$ értékeket, azaz fennáll $7 \cdot 36 + (-5) \cdot 50 = 2$
- megvizsgáljuk, hogy $d = 2$ osztja-e 1676-ot
- meghatározzuk:

$$x_0 = \hat{x} \cdot \frac{c}{d} = 7 \cdot \frac{1676}{2} = 7 \cdot 838 = 5866$$

$$y_0 = \hat{y} \cdot \frac{c}{d} = -5 \cdot \frac{1676}{2} = -5 \cdot 838 = -4190$$

- fennáll:

$$5866 \cdot 36 - 4190 \cdot 50 = 1676$$

Diofantoszi egyenletek

- Keressük a pozitív megoldásokat, fenn kell álljon:

$$5866 + n \cdot \frac{50}{d} = 5866 + n \cdot 25 \quad \geq 0 \quad \Rightarrow n \quad \geq -234.64$$

$$-4190 - n \cdot \frac{36}{d} = -4190 - n \cdot 18 \quad \geq 0 \quad \Rightarrow n \quad \leq -232.7$$

$$\begin{aligned} \Rightarrow \quad n &= -234 \\ x_1 &= 5866 + 25 \cdot (-234) = 16 \\ y_1 &= -4190 - 18 \cdot (-234) = 22 \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad n &= -233 \\ x_2 &= 5866 + 25 \cdot (-233) = 41 \\ y_2 &= -4190 - 18 \cdot (-233) = 4 \end{aligned}$$

Tehát:

- 1. megoldás: 16 láda almát és 22 láda körtét rendelt, $16 \cdot 36 + 22 \cdot 50 = 1676$.
- 2. megoldás: 41 láda almát és 4 láda körtét rendelt, $41 \cdot 36 + 4 \cdot 50 = 1676$.

Diofantoszi egyenletek

3. feladat

Írjunk egy Python függvényt, amely meghatározza egy kétismeretlenes diofantikus egyenlet pozitív megoldásait.

```
from math import ceil, floor
def diofant (a, b, c):
    (d, xk, yk) = extEuclid (a, b)
    if c % d != 0:
        print ('nincs megoldas')
        return
    x0 = xk * c//d
    y0 = yk * c//d
    bd, ad = b // d, a//d
    n1 = int(ceil(-x0 / bd))
    n2 = int(floor(y0 / ad))
    if n1 <= n2:
        print('megoldasok:')
        for i in range(n1, n2 + 1):
            print (x0 + bd * i, y0 - ad * i)
    else: print('nincs pozitiv megoldas')
```

```
>>> diofant(36, 50, 1676)
...
```

A strip, split függvények

a strip függvény levágja az adott String típusú adat elejéről és végéről a paraméterként megadott karaktert:

```
>>> 'Sapientia Egyetem\n\n'.strip()
    'Sapientia Egyetem'
>>> '!Sapientia Egyetem!!'.strip('!')
    'Sapientia Egyetem'
>>> 'Diszkret Matek\t\t'.strip('\t')
    'Diszkret Matek'
```

a split függvény az adott String típusú adatot feldarabolja a megadott karakter mentén, rész-Stringeket hozva létre:

```
>>> 'Diszkret Matek'.split()
    ['Diszkret', 'Matek']
>>> 'Diszkret\tMatek\tInformatika'.split('\t')
    ['Diszkret', 'Matek', 'Informatika']
>>> 'Diszkret Matek'.split('t')
    ['Diszkre', ' Ma', 'ek']
>>> 'Diszkret Matek'.split('te')
    ['Diszkret Ma', 'k']
```

A zip és a enumerate függvények

```
>>> hL =  ['januar','februar','marcius','aprilis']
>>> homL = [-3, -1, 3, 5]
>>> for h1, h2 in zip(hL, homL): # osszecipzarozzuk a ket listat
        print(h1, h2)

    januar -3
    februar -1
    marcius 3
    aprilis 5

>>> for i, h1 in enumerate(hL):
        print(i, h1)

    0 januar
    ...

>>> for i, (h1, h2) in enumerate(zip(hL, homL)):
        print(i, h1, h2)

    0 januar -3
    ...
```