Adatstruktúrák és Algoritmusok

3. gyakorlat

Rekurzív módon adott sorozatok, A mester tétel és alkalmazásai, Számelméleti algoritmusok I,

Rekurzív módon adott sorozatok

A Fibonacci sorozat

Leonardo Pisano (a pisai Leonardo), ismertebb nevén Fibonacci (Bonacci fia) (1170(?)-1240), olasz származású matematikus, tőle származik az úgynevezett ***Fibonacci sorozat****.*

A nyulak szaporodásának axiomatikus elmélete
I. Az első hónapban 1 pár nyulunk van.
II. Minden nyúlpár 2 hónapos korától havonta egy új nyúlpárnak ad életet.
III. A nyulak örök életűek.

Fibonacci kérdése: 1 év elteltével hány pár nyulunk lesz?

Megoldás

Jelölje $F(n)$ az n-edik hónapban a nyúlpárok számát. Ekkor a Fibonacci sorozat eleget tesz az

$$F(1) = 1, \qquad F(2) = 1$$
$$F(n+2) = F(n) + F(n+1) \qquad (n \geqslant 1)$$

kezdeti érték és rekurzió feltételeknek.
A rekurzió alapján könnyű kiszámolni a sorozat alábbi tagjait.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$F(n)$	0	1	1	2	3	5	8	13	21	34	55	89	144

Tehát válaszolva Fibonacci kérdésére 144 pár nyulunk lesz.
A rekurzív definíció felhasználásával az F_{100} kézzel történő kiszámolása már egy kicsit nehezebb.

A Binet formula

Az alábbi Binet formula mutatja a rekurzió feloldását, azaz az explicit alakot.

Binet formula

$$F_n = \frac{1}{\sqrt{5}}\left(\Phi^n - \overline{\Phi}^n\right), \qquad \textit{ahol} \quad \Phi = \frac{1+\sqrt{5}}{2}, \quad \Phi = \frac{1-\sqrt{5}}{2}$$

A rekurzív definíció és az explicit forma között az a különbség, hogy a rekurzív forma esetén, ha F_{100} kiszámítása lenne a cél, akkor szépen el kellene lépegetnünk egyesével 100-ig, az explicit összefüggés viszont "kapásból" megadja az eredményt.
A Binet formula alapján látható, hogy

$$F(100) = 3.5422 \cdot 10^{20},$$

azaz ha a nyulak valóban úgy szaporodnának, ahogyan azt a Fibonacci féle rekurzív formula előírja, akkor 100 hónap elteltével mindent elborítanának a nyulak.

F_n előállítása kerekítéssel

$$F(n) = \text{Round}\left(\frac{1}{\sqrt{5}}\Phi^n\right)$$

Feladat

Oldjuk fel az alábbi rekurziót: $F(1) = 1,\ T(n) = T\left(\frac{n}{2}\right) + 1$

- $T(1) = 1$
- $T(2) = T(1) + 1 = 1 + 1 = 2$
  $T(3) = -$
- $T(4) = T(2) + 1 = 2 + 1 = 3$
- $T(5) = -$
- $T(6) = T(3) + 1 = -$
- $T(7) = -$
- $T(8) = T(4) + 1 = 3 + 1 = 4$

Sejtés: $T(2^n) = n + 1 \quad \Rightarrow \quad T(n) = \log_2(n) + 1.$

A rekurzió feloldása

$T(n)$ *csak akkor számolható, ha* $n = 2^k$.
Definiáljuk az $U(k)$ *sorozatot* $U(k) := T(2^k)$ *módon. A rekurzió:*

$$U(k) = T(2^k) = T(2^{k-1}) + 1 = U(k-1) + 1 \qquad (k = 1, 2, \dots)$$

$U(0) = 1$, $U(1) = 2$. *Így* $U(k)$ *egy számtani sorozat* $U(1) = 2$, $d = 1$ *paraméterekkel.*
Ekkor

$$U(k) = U(1) + (k-1) \cdot 1 = 2 + (k-1)1 = k + 1.$$

A kapott összefüggés $k = 1$ *esetén is érvényes.*
Ekkor $T(2^k) = k + 1$. *Írjunk* k *helyére* $\log(n)$*-et, amiből kapjuk, hogy*

$$T(n) = \log(n) + 1$$

$n = 1, 2, \dots,$ *azaz* $T(n) = \Theta(\log(n))$.

A mester tétel és alkalmazása

Polinomiálisan lassabb és gyorsabb növekedés adott tesztfüggvénytől

Definíció *Legyen $p \geqslant 0$ és $f(n)$ egy növekedési függvény. Azt mondjuk, hogy*

- ***az $f(n)$ polinomiálisan lassabban nő, mint a n^p tesztpolinom**, ha $\exists \varepsilon > 0 : f(n) = \mathcal{O}(n^{p-\varepsilon})$*
- ***az $f(n)$ polinomiálisan gyorsabban nő, mint a n^p tesztpolinom**, ha $\exists \varepsilon > 0 : f(n) = \Omega(n^{p+\varepsilon})$*

Feladat

Bizonyítsuk be, hogy ha $\varepsilon > 0$, *akkor*

$$\log(n) = \mathcal{O}(n^{\varepsilon}),$$

így $\log(n)$ *polinomiálisan lassabban nő, mint az* n^{ε} *tesztpolinom.*

Megoldás

A L'Hospital szabály alkalmazásával kapjuk, hogy

$$\lim_{x\to\infty} \frac{\log(n)}{x^{\varepsilon}} = \lim_{x\to\infty} \frac{\frac{1}{\ln(2)}\frac{1}{x}}{\varepsilon x^{\varepsilon-1}} = \frac{1}{\ln(2)} \lim_{x\to\infty} \frac{1}{\varepsilon x^{\varepsilon}} = 0,$$

azaz $\log(n) = o(n^{\varepsilon})$, *amiből kapjuk, hogy* $\log(n) = \mathcal{O}(n^{\varepsilon})$.

Feladat

Legyenek $f(n) = n\log(n)$, $g(n) = n^p$ *valamely* $p \geqslant 0$ *esetén. Ekkor*

a. *Ha* $p > 1$, *akkor* $f(n) = n\log(n)$ *polinomiálisan lassabban nő, mint a* $g(n) = n^p$ *tesztpolinom.*

b. *Ha* $p = 1$, *akkor* $f(n) = n\log(n)$ *gyorsabban nő, mint a* $g(n) = n^p$ *tesztpolinom, de nem polinomiálisan.*

c. *Ha* $0 \leqslant p < 1$, *akkor* $f(n) = n\log(n)$ *polinomiálisan gyorsabban nő, mint a* $g(n) = n^p$ *tesztpolinom.*

A Mester tétel arra használható, hogy bizonyos speciális alakú rekurzív módon adott növekedési függvényről a rekurzió feloldása nélkül meg tudjuk mondani, hogy milyen növekedési rendű.

A mester tétel

Legyenek $a \geqslant 1$, $b > 1$, $p = \log_b(a)$, $a, b \in \mathbb{R}$, $f : \mathbb{Z}_+ \to \mathbb{R}_+$ *egy növekedési függvény,* $g(n) = n^p$ *a tesztpolinom.*

Rekurziós összefüggés:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

M1 *Ha* $f(n)$ *polinomiálisan lassabb növekedésű, mint a* $g(n) = n^p$ *tesztpolinom, akkor* $T(n) = \Theta(g(n))$.

M2 *Ha* $f(n) = \Theta(g(n))$, *akkor* $T(n) = \Theta(g(n)\log(n))$.

M3 *Ha* $f(n)$ *polinomiálisan gyorsabb növekedésű, mint a* $g(n) = n^p$ *tesztpolinom és teljesül az úgynevezett* ***regularitási feltétel****, azaz*

$$\exists(c < 1)\exists(n_0 \in \mathbb{Z}_+)\forall(n \geqslant n_0) : af\left(\frac{n}{b}\right) \leqslant cf(n)$$

akkor $T(n) = \Theta(f(n))$.

Feladatok

1. $T(n) = T(\frac{n}{2}) + 1$.
   $a = 1$, $b = 2$, $p = \log_2(1) = 0$, $f(n) = n^0 = 1$, $g(n) = n^0 = 1$.
   Ekkor $f(n) = \Theta(g(n))$. Így a $M2$ alapján $T(n) = \Theta(1. \log(n))$.
2. $T(n) = 4T(\frac{n}{2}) + n$.
   $a = 4$, $b = 2$, $p = \log_b(a) = \log_2(4) = 2$, $f(n) = n^1$,
   $g(n) = n^2$. Az $f(n) = n^1$ polinomiálisan lassabban nő, mint
   $g(n) = n_2$ tesztpolinom, így a $M1$ alapján
   $T(n) = \Theta(g(n)) = \Theta(n^2)$.
3. $T(n) = 4T(\frac{n}{2}) + n^2$.
   $a = 4$, $b = 2$, $p = \log_b(a) = \log_2(4) = 2$, $f(n) = n^2$,
   $g(n) = n^2$. $f(n) = \Theta(g(n))$, $M2$ alapján
   $T(n) = \Theta(g(n) \log(n)) = \Theta(n^2 \log(n))$.

4. $T(n) = 4T(\frac{n}{3}) + n^3$.
$a = 4$, $b = 3$, $p = \log_b(a) = \log_3(4)$, $f(n) = n^3$, $g(n) = n^p$.

$$p = \log_3(4) < \log_3(27) = 3$$

(itt használtunk egy kis becslést, de elég lett volna $\log_3(4)$-et beütni a számológépbe ahhoz, hogy megkapjuk, hogy $\log_3(4) < 3$).
$f(n) = n^3$ polinomiálisan gyorsabban nő, mint a $g(n)$ tesztpolinom. Az $M3$-hoz még ellenőrizni kell még a relgulatitási feltételt:

$$af\left(\frac{n}{b}\right) = 4 \cdot \left(\frac{n}{3}\right)^3 = \frac{4}{27} \cdot n^3 = \frac{4}{27} \cdot f(n), \quad c = \frac{4}{27} < 1.$$

Így kapjuk, hogy $T(n) = \Theta(f(n)) = \Theta(n^3)$.

5. $T(n) = 3T(\frac{n}{4}) + n\log(n)$.
$a = 3$, $b = 4$, $p = \log_b(a) = \log_4(3)$, $f(n) = n\log(n)$,
$g(n) = n^p$

$$p = \log_4(3) < \log_4(4) = 1$$

Ekkor $f(n) = n\log(n)$ polinomiálisan gyorsabban nő, mint a $g(n) = n^{\log_4(3)}$.
Az $M3$-hoz ellenőrizni kell még a regularitási feltételt.

$$af\left(\frac{n}{b}\right) = 3\frac{n}{4}\log\left(\frac{n}{4}\right) < \frac{3}{4}n\log(n) = \frac{3}{4}f(n), \quad c = \frac{3}{4} < 1$$

Így $T(n) = \Theta(f(n)) = \Theta(n\log(n))$.

6. $T(n) = 2T(\frac{n}{2}) + n\log(n)$
   $a = 2$, $b = 2$, $p = \log_b(a) = \log_2(2) = 1$, $f(n) = n\log(n)$, $g(n) = n^p = n^1$. Ekkor $f(n)$ gyorsabban nő, mint a $g(n)$, de nem polinomálisan, így **a Mester tétel nem alkalmazható**.

Számelméleti algoritmusok I

Számelméleti definíciókat (fogalmak) és tételeket (állítások) a $\mathbb{Z}$ halmazon fogalmazzuk meg, az algoritmusok $\mathbb{Z}_+ \cup \{0\}$ halmazon futtatjuk.

Oszthatósági reláció $\mathbb{Z}$-n

Definíció *Legyenek $a, b \in \mathbb{Z}$. "a" **osztója** "b"-nek, ha létezik $c \in \mathbb{Z}$ úgy, hogy $b = ac$. Jele: $a|b$.*

Prímszám

Definíció *Egy $p \in \mathbb{Z}_+$, $p > 1$ számot **prímszám**nak nevezünk, ha 1-en és önmagán kívül nincs más pozitív osztója.*

Megjegyzés

A prímszám szokásos definíciója: egy p 0-tól és ± 1-től különböző egész számot prímszámnak nevezünk, ha valhányszor osztója egy szorzatnak, mindannyiszor osztója a szorzat valamelyik tényezőjének, azaz

$$p|ab \qquad \Longleftrightarrow \qquad p|a \text{ vagy } p|b.$$

Mi ezt a tulajdonságot a későbbiekben ***prímtulajdonság****nak fogjuk nevezni.*
Egy p 0-tól és ± 1-től különböző egész számot felbonthatatlannak, vagy ***irreducibilis****nek nevezünk, ha nincs valódi faktorizációja, azaz $p = ab$-ből következik, hogy $a = \pm 1$, vagy $a = \pm p$.*
Bizonyítható, hogy az egész számok gyűrűjében a prímszámok és az irreducibilis számok egybeesnek.

Legnagyobb közös osztó

Definíció *Az a, $b \in \mathbb{Z}$ számok* ***legnagyobb közös osztója****:*

$$\operatorname{lnko}(a,b) = \begin{cases} 0, & \text{ha } a = 0 \text{ és } b = 0 \\ \max\{d \in \mathbb{Z}_+ \mid d|a \text{ és } d|b\} & \text{egyébként} \end{cases}$$

Megjegyzés

Ha az a és b számok közül nem mindkettő 0, akkor egy δ számot az a és b számok ***kitüntetett közös osztó****jának nevezünk, ha a δ szám*

1. *az a és b számok közös osztója;*
2. *az a és b számok bármely közös osztójának a többszöröse.*

Elemi számelméletben meg szokták mutatni, hogy ha az a és b számok közül nem mindkettő 0, akkor az a és b számok legnagyobb közös osztója egyúttal a kitüntetett közös osztójuk is.

Jelölés

Tetszőleges $a,\ b \in \mathbb{Z}$ *számok esetén legyen*

$$L(a,b) := \{ua + vb \mid u \in\in \mathbb{Z},\ v \in \mathbb{Z}\}.$$

Az $L(a,b)$ *halmazt az* a ***és*** b ***elemekből képzett lineáris kombinációk halmazá****nak nevezzük.*

Most megismerkedünk a reprezentációs, a redukciós és a rekurziós tétellel.

Fontos tételek

1. **Reprezentációs tétel** *Ha a és b egyidejűleg nem 0, akkor*

$$\mathrm{lnko}(a, b) = \min\{x \in L(a, b) \mid x > 0\}.$$

Következmény *A legnagyobb közös osztó bármely közös osztó többszöröse.*

2. **Redukciós tétel** *Legyenek $a, b \in \mathbb{Z}$. Ekkor*

$$\mathrm{lnko}(a, b) = \mathrm{lnko}(a - b, b)$$

3. **Rekurziós tétel** *Legyenek $a, b \in \mathbb{Z}$. Ekkor*

$$\mathrm{lnko}(a, b) = \mathrm{lnko}(b, a \bmod b)$$

Redukciós algoritmus

	REDUK(a,b $\parallel$ d^*)
INPUT	$a, b \in \mathbb{Z}_+ \cup \{0\}$
OUTPUT	d^*
1.	WHILE $(b \neq 0)$
2.	IF $b > a$ THEN CSERE(a, b)
3.	$a \longleftarrow a - b$
4.	$d^* \longleftarrow a$
5.	RETURN(d^*)

Euklideszi algoritmus

	EUK(a,b $\parallel$ d^*)
INPUT	$a, b \in \mathbb{Z}_+ \cup \{0\}$
OUTPUT	d^*
1.	WHILE $(b \neq 0)$
2.	$\binom{a}{b} \longleftarrow \binom{b}{a \mod b}$
3.	$d^* \longleftarrow a$
4.	RETURN(d^*)

Példa

Határozzuk meg a redukciós algoritmus és a rekurziós algoritmus segítségével a 90 és a 24 legnagyobb közös osztóját. A bulet módon megjelölt lépések azok, amelyeken keresztül a rekurziós algoritmus eljut a legnagyobb közös osztóhoz.

$$\dot{\text{lnko}}\ (90, 24) = \text{lnko}(66, 24) = \text{lnko}(42, 24) = \text{lnko}(18, 24) =$$
$$= \dot{\text{lnko}}\ (24, 18) = \text{lnko}(6, 18) = \dot{\text{lnko}}\ (18, 6) =$$
$$= \text{lnko}(12, 6) = \text{lnko}(6, 6) = \text{lnko}(0, 6) =$$
$$= \dot{\text{lnko}}\ (6, 0) = 6$$

A rekurziós algoritmus, mint Euklideszi osztások sorozata

A rekurziós algoritmust lehet úgy értelmezni, mint Euklideszi osztások sorozatát, melyben

- Minden egyes lépésben az osztóból lesz az osztandó és a maradékból lesz az osztó;
- Ezt az eljárást addig iteráljuk, amíg a maradék nem válik nullává;
- A legnagyobb közös osztó a maradékok sorozatában az utolsó zérustól különböző maradék.

$$
\begin{aligned}
a_0 &= q_0 b_0 + r_0 \\
b_0 &= q_1 r_0 + r_1 \\
r_0 &= q_2 r_1 + r_2 \\
&\vdots \\
r_{n-2} &= q_n r_{n-1} + \boxed{r_n} \\
r_{n-1} &= q_{n+1} r_n + 0
\end{aligned}
$$

Lamé tétele

Ha az euklideszi algoritmusban $a > b \geqslant 0$ *és* $F_{k+1} > b$, *akkor a rekurzív hívások száma kevesebb, mint* k.

KÖSZÖNÖM A FIGYELMET!