Haladó informatikai algoritmusok

Járdakövezés

Dinamikus programozás jellemzői

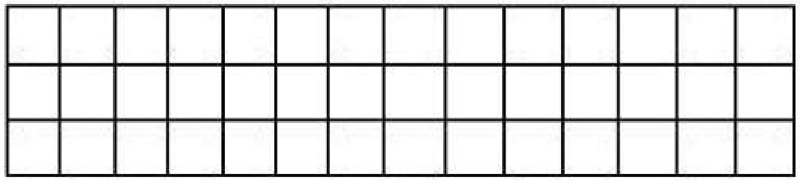
Egy dinamikus programozási algoritmus kifejlesztése 4 lépésre bontható fel:

1. Jellemezzük az optimális megoldás szerkezetét.
2. Rekurzív módon definiáljuk az optimális megoldás értékét.
3. Kiszámítjuk az optimális megoldás értékét alulról felfelé történő módon.
4. A kiszámított információk alapján megszerkesztünk egy optimális megoldást.

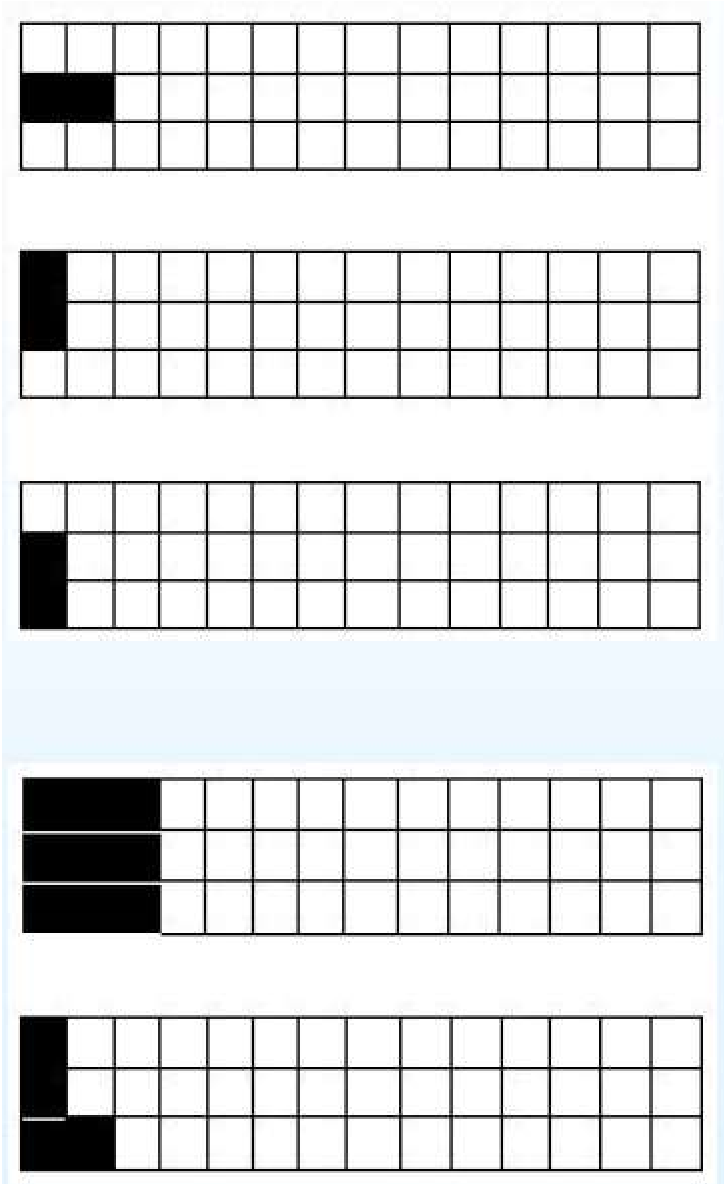
Járdakövezés problémája

Számítsuk ki, hogy hány féle képpen lehet egy $3 \times n$ **egység méretű járdát kikövezni** 1×2 méretű lapokkal!
Jelölje $A(n)$ a megoldás értékét $3 \times n$ méretű járda esetén.
Az könnyen megállapítható, hogy az első oszlop középső négyzete 3 féle képpen fedhető le.
Jelölje $B(n)$ azt, hogy hány féle képpen fedhető le egy $3 \times n$ méretű járda, amelynek a bal alsó sarka már le van fedve.
Szimmetria miatt a jobb felső sarok lefedettsége esetén is $B(n)$-féle lefedés van.

Hogyan tudjuk felírni $A(n)$ és $B(n)$ meghatározását egy-egy rekurzív összefüggéssel?



Különböző esetek:



Rekurzív összefüggések

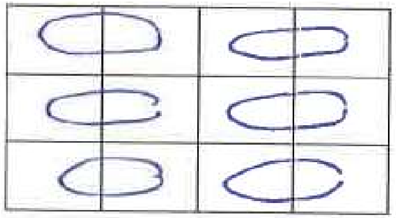
$A(n)$ **számítása:**

$$A(n) = \begin{cases} 0, & \text{ha } n = 1 \\ 3, & \text{ha } n = 2 \\ A(n-2) + 2 * B(n-1), & \text{ha } n > 2 \end{cases}$$

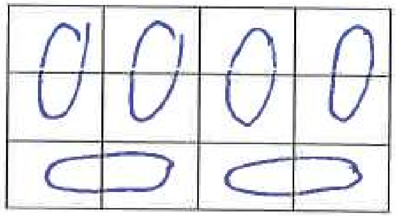
$B(n)$ **számítása:**

$$B(n) = \begin{cases} 1, & \text{ha } n = 1 \\ 0, & \text{ha } n = 2 \\ A(n-1) + B(n-2), & \text{ha } n > 2 \end{cases}$$

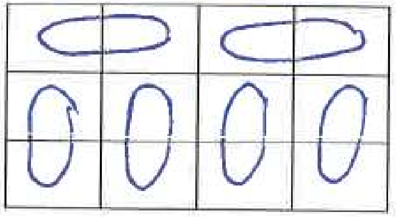
1.



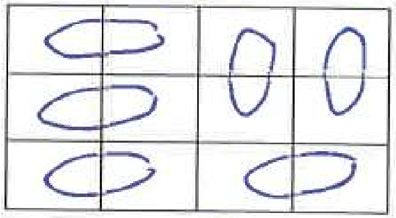
2.



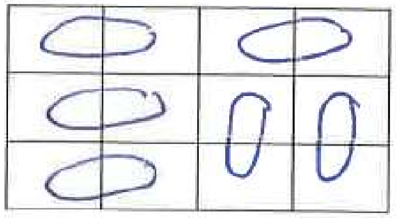
3.



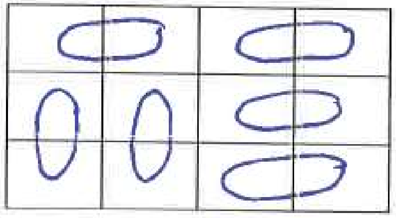
4.



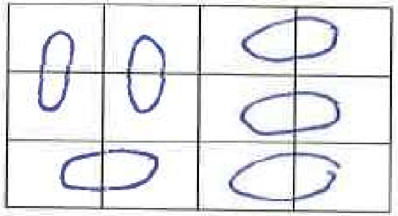
5.



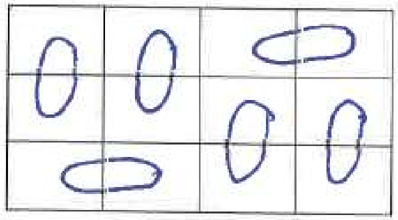
6.



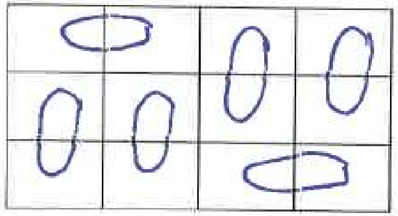
7.



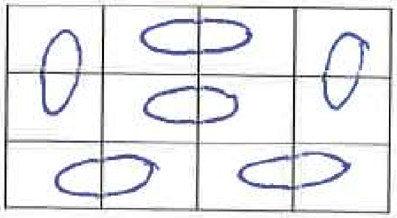
8.



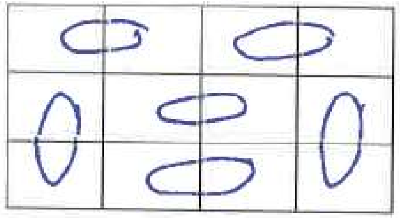
9.



10.



11.

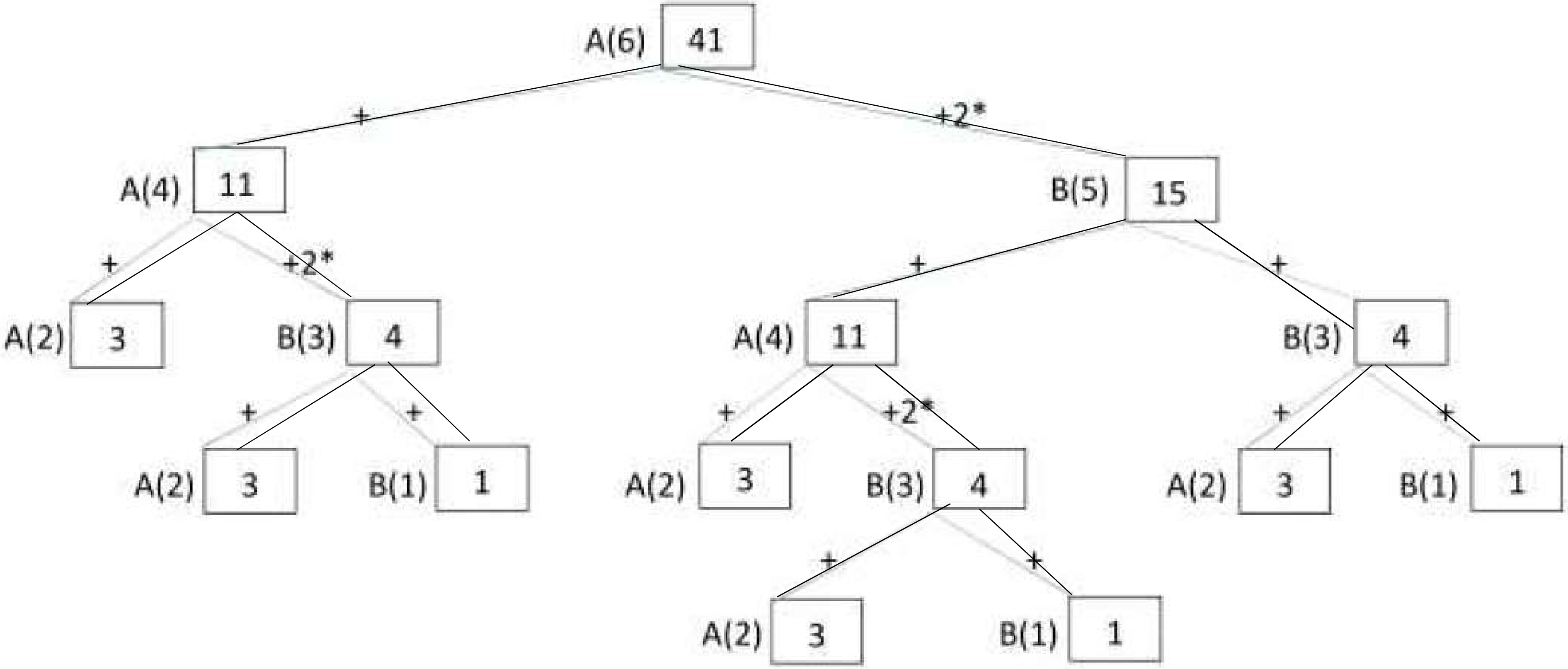


$$A(4) = \underbrace{A(2)}_{3} + 2\cdot \underbrace{B(3)}_{2\cdot\left(\underbrace{A(2)}_{3} + \underbrace{B(1)}_{1}\right)} = 11$$

Rekurziós fa

Számolja ki n=6 esetén a lehetséges esetek számát!

A rekurzív képletek segítségével felrajzolható egy rekurziós fa:



Táblázat kitöltés

Megoldás dinamikus programozási módszerrel, táblázat kitöltéssel:

```
program jarda;
const maxN=100;
var A,B: array[1..maxN] of Int64;
n,i: integer;
begin
n:=64;
A[1]:=0; A[2]:=3;
B[1]:=1; B[2]:=0;
for i:=3 to n do begin
   A[i]:=A[i-2]+2*B[i-1];
   B[i]:=A[i-1]+B[i-2];
end;
writeln('A(',n,') = ',A[n]);
end.
```

	1.	2.	3.	4.	5.	6.	7.	8.	9.
A	0	3	0	11	0	41	0	153	0
B	1	0	4	0	15	0	56	0	209

Haladó informatikai algoritmusok

Nyomtatási feladat

Nyomtatási feladat

Feladat az $s_1, s_2, \ldots, s_n$ szavakból álló bekezdés kinyomtatása.
A szavak rendre $l_1, l_2, \ldots, l_n$ karakterekből állnak.
A nyomtató egy sorba összesen M karaktert tud elhelyezni.
Tegyük fel, hogy $l_i \leq M \quad \forall\, 1 \leq i \leq n$ esetén.
Ha egy sor az s_i-től s_j-ig terjedő szavakat tartalmazza, akkor a szavak között mindig egy szóköz van, míg a sor végén további

$$M - j + i - \sum_{m=i}^{j} l_m$$

extra szóköz (nem negatív!).
Adjunk meg egy hatékony algoritmust, amely a "legszebben" nyomtatja ki a bekezdést, azaz minimalizálja az utolsó sor kivételével a sorok végén található extra szóközök számának köbeinek összegét!

Megoldás dinamikus programozással

$\forall 1 \leq i \leq j \leq n$ esetén legyen

$$e[i,j] = M - j + i - \sum_{m=i}^{j} l_m$$

(az s_i-től s_j-ig terjedő szavakat tartalmazó sor végén lévő extra szóközök száma)
$\forall 1 \leq i \leq j \leq n$ esetén legyen

$$k[i,j] \quad = \quad \begin{cases} \infty, \text{ ha } e[i,j] < 0, \\ 0, \text{ ha } j = n \text{ és } e[i,j] \geq 0, \\ e[i,j]^3, \text{ különben} \end{cases}$$

Megoldás dinamikus programozással

Tekintsük az $s_1, s_2, \ldots, s_j$ szavak egy optimális elrendezését.
Tegyük fel, hogy ebben az utolsó sor az s_i szóval kezdődik.
Vegyük észre, hogy ekkor az utolsó előtti sorig terjedő rész az $s_1, s_2, \ldots, s_{i-1}$ szavak egy optimális elrendezése.
Ha nem így lenne, akkor ezt a részt lecserélve az $s_1, s_2, \ldots, s_{i-1}$ szavak egy optimális elrendezésére az $s_1, s_2, \ldots, s_j$ szavak egy "szebb" elrendezéséhez jutnánk, ami ellentmondás.
Jelölje $c[j]$ az $s_1, s_2, \ldots, s_j$ szavak optimális elrendezésének soraihoz rendelt k értékek összegét.

$$c[j] = \begin{cases} 0, \text{ ha } j = 0, \\ min\{c[i-1] + k[i,j] \mid 1 \leq i \leq j\}, \text{ ha } j > 0. \end{cases}$$

$\forall\, 1 \leq i \leq j$ estén legyen $p[j]$ annak az s_i szónak az indexe, amellyel az utolsó sor kezdődik az $s_1, s_2, \ldots, s_j$ szavak optimális elrendezésénél.

Program

```
Nyomtatás(l,n,M)
for i = 1 to n do
   e[i,i] = M − l[i]
   for j = i + 1 to n do
   e[i,j] = e[i,j − 1] − l[j] − 1
for i = 1 to n do
   for j = i to n do
      if e[i,j] < 0
        then k[i,j] = ∞
        else
          if j = n and e[i,j] ≥ 0
            then k[i,j] = 0
            else k[i,j] = e[i,j]^3
c[0] = 0
for j = 1 to n do
  c[j] = ∞
  for i = 1 to j do
    if c[i − 1] + k[i,j] < c[j] then
       c[j] = c[i − 1] + k[i,j]; p[j] = i
return c,p
```

Tördelés

Tördelés(p,j)
$i = p[j]$
if $i = 1$
    then $h = 1$
    else $h = Trdels(p, i-1) + 1$
print (h, i, j)
return h

Az eljárás az optimálisan tördelt bekezdés sorainak számával tér vissza.
A kinyomtatott (h, i, j) hármasok azt mutatják, hogy a h-adik sor az s_i-től s_j-ig terjedő szavakat tartalmazza ($h = 1, 2, \ldots$).

A Nyomtatás eljárás költsége $O(n^2)$, a Tördelés eljárás költsége $O(n)$.

Példa:

s_1	s_2	s_3	s_4	s_5	s_6	s_7
Még	nyílnak	a	völgyben	a	kerti	virágok ...
$l_1=3$	$l_2=7$	$l_3=1$	$l_4=8$	$l_5=1$	$l_6=5$	$l_7=8$

...

e	1	2	3	4	5	6	7
1	9	1	−1	—	—	—	—
2		5	3	−6	—	—	—
3			11	2	0	—	—
4				4	5	−4	—
5					11	5	−4
6						7	−2
7							4

$M = 12$

$$e[i,j] = 12 - j + i - \sum_{m=i}^{j} l_m$$

$n = 8$

$$k[i,j]=\begin{cases}\infty, & \text{ha } e[i,j]<0\\ 0, & \text{ha } j=n \text{ és } e[i,j]\ge 0\\ e[i,j]^3, & \text{különben}\end{cases}$$

k	1	2	3	4	5	6	7
1	9^3	1	∞	∞	∞	∞	∞
2		125	27	∞	∞	∞	∞
3			11^3	8	0	∞	∞
4				64	125	∞	∞
5					11^3	125	∞
6						7^3	∞
7							0

$$C\ \begin{array}{cccccccc} 0. & 1. & 2. & 3. & 4. & 5. & 6. & 7. \\ \end{array}$$
$$\left[\begin{array}{cccccccc} 0 & \cancel{\infty} & \cancel{\infty} & \cancel{\infty} & \cancel{\infty} & \cancel{\infty} & \cancel{\infty} & \infty \\ & 9^3 & 1 & 28 & 36 & 28 & 153 & 153 \end{array}\right]$$

$$P\ \begin{array}{ccccccc} 1. & 2. & 3. & 4. & 5. & 6. & 7. \end{array}$$
$$\left[\begin{array}{ccccccc} 1 & 1 & 2 & 3 & 3 & 5 & 7 \end{array}\right]$$

$\boxed{j=1,\ i=1}$

$c[0]+k[1,1] \stackrel{?}{<} c[1]$

$0+9^3<\infty$ ✓

$c[1]=c[0]+k[1,1]=$

$=0+9^3$ $\quad p[1]=1$

$\boxed{j=2,\ i=1}$

$c[0]+k[1,2] \stackrel{?}{<} c[2]$

$0+1<\infty$ ✓

$c[2]=c[0]+k[1,2]=1$

$p[2]=1$

$\boxed{j=2,\ i=2}$

$c[1]+k[2,2] \stackrel{?}{<} c[2]$

$9^3+5^3<1$ ✗

$\boxed{j=3,\ i=1}$

$c[0]+k[1,3] \stackrel{?}{<} c[3]$

$0+\infty<\infty$ ✗

$\boxed{j=3,\ i=2}$

$c[1]+k[2,3] \stackrel{?}{<} c[3]$

$9^3+3^3<\infty$ ✓

$c[3]=c[2]+k[2,3]=1+27=28$

$p[3]=2$

$\boxed{j=3,\ i=3}$

$c[2]+k[3,3] \stackrel{?}{<} c[3]$

$1+11^3<28$ ✗

$\boxed{j=4,\ i=1}$

$c[0]+k[1,4] \stackrel{?}{<} c[4]$

$0+\infty<\infty$ ✗

$\boxed{j=4,\ i=2}$

$c[1]+k[2,4] \stackrel{?}{<} c[4]$

$9^3+\infty<\infty$ ✗

$\boxed{j=4,\ i=3}$

$c[2]+k[3,4] \stackrel{?}{<} c[4]$

$1+8<\infty$ ✓

$c[4]=c[3]+k[3,4]=$

$=28+8=36$ $\quad p[4]=3$

$\boxed{j=4,\ i=4}$

$c[3]+k[4,4] \stackrel{?}{<} c[4]$

$28+64<36$ ✗

$\boxed{j=5,\ i=1}$

$c[0]+k[1,5]<c[5]$

$0+\infty<\infty$ ✗

$\boxed{j=5,\ i=2}$

$c[1]+k[2,5] \stackrel{?}{<} c[5]$

$9^3+\infty<\infty$ ✗

$\boxed{j=5,\ i=3}$

$c[2]+k[3,5] \stackrel{?}{<} c[5]$

$1+0<\infty$ ✓

$c[5]=c[3]+k[3,5]=$

$=28+0=28$ $\quad p[5]=3$

$\boxed{j=5,\ i=4}$

$c[3]+k[4,5] \stackrel{?}{<} c[5]$

$28+125<28$ ✗

$\boxed{j=5,\ i=5}$

$c[4]+k[5,5] \stackrel{?}{<} c[5]$

$36+11^3<28$ ✗

$\boxed{j=6,\ i=1}$

$c[0]+k[1,6] \stackrel{?}{<} c[6]$

$0+\infty<\infty$ ✗

$\boxed{j=6,\ i=2}$

$c[1]+k[2,6] \stackrel{?}{<} c[6]$

$9^3+\infty<\infty$ ✗

$\boxed{j=6,\ i=3}$

$c[2]+k[3,6] \stackrel{?}{<} c[6]$

$1+\infty<\infty$ ✗

$\boxed{j=6,\ i=4}$

$c[3]+k[4,6] \stackrel{?}{<} c[6]$

$28+\infty<\infty$ ✗

$\boxed{j=6,\ i=5}$

$c[4]+k[5,6] \stackrel{?}{<} c[6]$

$36+125<\infty$ ✓

$c[6]=c[5]+k[5,6]=$

$=28+125=153$

$p[6]=5$

$\boxed{j=6,\ i=6}$

$c[5]+k[6,6] \stackrel{?}{<} c[6]$

$28+7^3<153$ ✗

$\boxed{j=7,\ i=1}$

$c[0]+k[1,7] \stackrel{?}{<} c[7]$

$0+\infty<\infty$ ✗

$\boxed{j=7,\ i=2}$

$c[1]+k[2,7] \stackrel{?}{<} c[7]$

$9^3+\infty<\infty$ ✗

$\boxed{j=7,\ i=3}$

$c[2]+k[3,7] \stackrel{?}{<} c[7]$

$1+\infty<\infty$ ✗

$\boxed{j=7,\ i=4}$

$c[3]+k[4,7] \stackrel{?}{<} c[7]$

$28+\infty<\infty$ ✗

$\boxed{j=7,\ i=5}$

$c[4]+k[5,7] \stackrel{?}{<} c[7]$

$36+\infty<\infty$ ✗

$\boxed{j=7,\ i=6}$

$c[5]+k[6,7] \stackrel{?}{<} c[7]$

$28+\infty<\infty$ ✗

$\boxed{j=7,\ i=7}$

$c[6]+k[7,7] \stackrel{?}{<} c[7]$

$153+0<\infty$ ✓

$c[7]=c[6]+k[7,7]=153$

$p[7]=7$

$$p = \begin{array}{ccccccc} 1. & 2. & 3. & 4. & 5. & 6. & 7. \\ \hline 1, & 1, & 2, & 3, & 3, & 5, & 7 \end{array}$$

Tördelés(p, 7)

$i := p[7] = 7$

$i \overset{?}{=} 1$ ✗

$h = \text{Tördelés}(p, 6) + 1 \Rightarrow 4$ $\quad\boxed{4, 7, 7}$

h i j

$h=3$

$i = p[6] = 5$

$i \overset{?}{=} 1$ ✗

$h = \text{Tördelés}(p, 4) + 1 \Rightarrow 3$ $\quad\boxed{3, 5, 6}$

h i j

$h=2$

$i = p[4] = 3$

$i \overset{?}{=} 1$ ✗

$h = \text{Tördelés}(p, 2) + 1 \Rightarrow 2$ $\quad\boxed{2, 3, 4}$

h i j

$i = p[2] = 1$

$i \overset{?}{=} 1$ ✓ $\quad h = 1$

$\boxed{1, 1, 2}$

h i j

$h=1$

sor ↓ meltől ↓ meddig ↓

$\boxed{1, 1, 2}$	⇒ Még␣nyílnak␣	12
$\boxed{2, 3, 4}$	⇒ a␣völgyben␣␣	12
$\boxed{3, 5, 6}$	⇒ a␣kerti␣␣␣␣␣	12
$\boxed{4, 7, 7}$	⇒ virágok␣␣␣␣␣	12