



ELTE | IK

PROGRAMOZÁS

Mintamegoldás

Horváth Győző



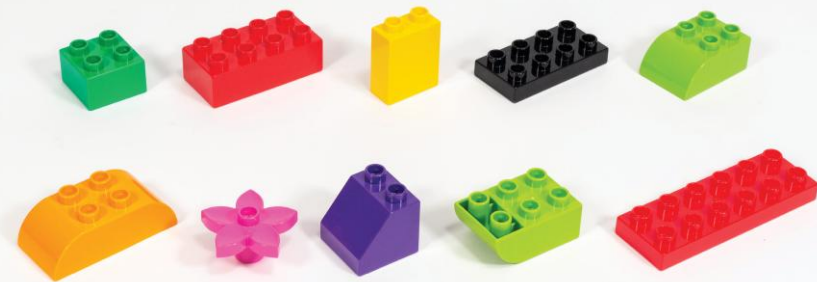
Ismétlés



Programozási minták

1. Összegzés
2. Megszámolás
3. Maximumkiválasztás
 - a. Minimumkiválasztás
4. Feltételes maximumkeresés
5. Keresés
6. Eldöntés
 - a. Mind eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás

Most Common DUPLO Parts



Brick Architect 

Programozási minták

Összegzés

i	f(i)
e	→ f(e)
e+1	→ f(e+1)
e+2	→ f(e+2)
...	→ ...
u-2	→ f(u-2)
u-1	→ f(u-1)
u	→ f(u)
=	
s	

Megszámolás

i	T(i)	érték
e	→ IGAZ	1
e+1	→ HAMIS	0
e+2	→ HAMIS	0
...	→
u-2	→ IGAZ	1
u-1	→ IGAZ	1
u	→ HAMIS	0
=		
db		

Maximum kiválasztás

i	f(i)
e	→ f(e)
e+1	→ f(e+1)
e+2	→ f(e+2)
...	→ ...
u-2	→ f(u-2)
u-1	→ f(u-1)
u	→ f(u)
maxind, maxért	

Feltételes maximumkeresés

i	T(i)	f(i)
e	→ HAMIS	f(e)
e+1	→ IGAZ	f(e+1)
e+2	→ IGAZ	f(e+2)
...	→
u-2	→ HAMIS	f(u-2)
u-1	→ IGAZ	f(u-1)
u	→ HAMIS	f(u)
van, maxind, maxért		

Programozási minták

Keresés

i	T(i)
e	→ HAMIS
e+1	→ HAMIS
e+2	→ IGAZ
...	→ ...
u-2	→ IGAZ
u-1	→ IGAZ
u	→ HAMIS
van, ind	

Eldöntés

i	T(i)
e	→ HAMIS
e+1	→ HAMIS
e+2	→ IGAZ
...	→ ...
u-2	→ IGAZ
u-1	→ IGAZ
u	→ HAMIS
van	

Kiválasztás

i	T(i)
e	→ HAMIS
e+1	→ HAMIS
e+2	→ IGAZ
...	→ ...
u-2	→ IGAZ
u-1	→ IGAZ
u	→ HAMIS
ind	

Programozási minták

Másolás

i			f(i)
e	→	1	f(e)
e+1	→	2	f(e+1)
e+2	→	3	f(e+2)
...	→
u-2	→	u-e-1	f(u-2)
u-1	→	u-e	f(u-1)
u	→	u-e+1	f(u)

y

Kiválogatás

i	T(i)	f(i)		y
e	→ HAMIS	f(e)	1	f(e+1)
e+1	→ IGAZ	f(e+1)	2	f(e+2)
e+2	→ IGAZ	f(e+2)	db= 3	f(u-1)
...	→	...		
u-2	→ HAMIS	f(u-2)		
u-1	→ IGAZ	f(u-1)		
u	→ HAMIS	f(u)		

db, y

Feladatmegoldási minta



Feladatmegoldási minta gyorsabb vonat az előzőnél

Feladat a Mesterről

Gyorsabb vonat az előzőnél

Ismerjük N vonat menetidejét Budapestről Siófokra.

Írj programot, amely megad egy vonatot, amely gyorsabb, mint az előző!

Bemenet

A *standard bemenet* első sorában a vonatok száma van ($1 \leq N \leq 100$). A következő N sor mindegyike egy-egy egész számot tartalmaz, az egyes vonatok menetidejét ($1 \leq M \leq 1000$).

Kimenet

A *standard kimenet* első sorába egy az előzőnél gyorsabb vonat sorszámát kell írni (ha több ilyen is van, akkor az első)! Ha nincs ilyen vonat, akkor -1-et kell írni!

Példa

Bemenet

6
118
200
199
116
200
122

Kimenet

3



Feladatmegoldási minta gyorsabb vonat az előzőnél

Biztosan van ilyen vonat? Ha igen,
akkor melyik az?

→ keresés: adott tulajdonságú elem
létezése és helye

Feladat:

Adj meg egy előzőnél gyorsabb vonatot!

Specifikáció:

Be: $n \in \mathbb{N}$, $\text{midő} \in [1..n]$

Ki: $\text{van} \in \mathbb{L}$, $\text{melyik} \in \mathbb{N}$

Ef: -

Uf: $(\text{van}, \text{melyik}) = \text{KERES}(i=2..n, \text{midő}[i] < \text{midő}[i-1])$

	6
1	118
2	200
3	199
4	116
5	200
6	122

1. Mik az intervallum határai? (2..6)
2. Milyen tulajdonságot vizsgálunk az intervallum egyes pontján?
3. Milyen néven tároljuk a keresés eredményeit?

i	T(i)
e	→ HAMIS
e+1	→ HAMIS
e+2	→ IGAZ
...	→ ...
u-2	→ IGAZ
u-1	→ IGAZ
u	→ HAMIS
van, ind	

Feladatmegoldási minta

gyorsabb vonat az előzőnél

Feladatsablon

(mintafeladat)

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $\text{van} \in \mathbb{L}$, $\text{ind} \in \mathbb{Z}$

Ef: -

Uf: $(\text{van}, \text{ind}) = \text{KERES}(i = e..u,$
 $\quad T(i))$

$\text{ind} \sim \text{melyik}$

$e..u \sim 2..n$

$T(i) \sim \text{midő}[i] < \text{midő}[i-1]$

$\text{ind} := e$

$\text{ind} \leq u$ és nem $T(\text{ind})$

$\text{ind} := \text{ind} + 1$

$\text{van} := \text{ind} \leq u$

Előzőnél gyorsabb vonat

(konkrét feladat)

Be: $n \in \mathbb{N}$, $\text{midő} \in \mathbb{N}[1..n]$

Ki: $\text{van} \in \mathbb{L}$, $\text{melyik} \in \mathbb{N}$

Ef: -

Uf: $(\text{van}, \text{melyik}) = \text{KERES}(i = 2..n,$
 $\quad \text{midő}[i] < \text{midő}[i-1])$

$\text{melyik} := 2$

$\text{melyik} \leq n$ és

nem $\text{midő}[\text{melyik}] < \text{midő}[\text{melyik} - 1]$

$\text{melyik} := \text{melyik} + 1$

$\text{van} := \text{melyik} \leq n$



Feladatmegoldási minta

gyorsabb vonat az előzőnél

Kódolás alapsablona

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be,ki)
```

```
// Beolvasás (specifikáció be)
```

```
// Feldolgozás (algoritmus, stuki)
```

```
// Kiírás (specifikáció ki)
```

}

Feladatmegoldási minta

gyorsabb vonat az előzőnél

Deklarálás

```
static void Main(string[] args) {
    // Deklarálás (változók, specifikáció be,ki)
    int n;
    int[] mido;
    bool van;
    int melyik;
    // Beolvasás (specifikáció be)

    // Feldolgozás (algoritmus, stuki)

    // Kiírás (specifikáció ki)
}
```

Feladatmegoldási minta gyorsabb vonat az előzőnél

Beolvasás

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be,ki)  
    int n;  
    int[] mido;  
    bool van;  
    int melyik;  
    // Beolvasás (specifikáció be)  
    Console.Write("n = ");  
    int.TryParse(Console.ReadLine(), out n);  
    mido = new int[n];  
    for (int i = 1; i <= n; i++) {  
        Console.Write("{0}. menetido = ", i);  
        int.TryParse(Console.ReadLine(), out mido[i - 1]);  
    }  
    // Feldolgozás (algoritmus, stuki)  
  
    // Kiírás (specifikáció ki)
```

Be: $n \in \mathbb{N}$,

$\text{midő} \in \mathbb{N}[1..n]$

```
}
```

Feladatmegoldási minta

gyorsabb vonat az előzőnél

Feldolgozás

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be,ki)  
    int n;  
    int[] mido;  
    bool van;  
    int melyik;  
    // Beolvasás (specifikáció be)  
    Console.Write("n = ");  
    int.TryParse(Console.ReadLine(), out n);  
    mido = new int[n];  
    for (int i = 1; i <= n; i++) {  
        Console.Write("{0}. menetido = ", i);  
        int.TryParse(Console.ReadLine(), out mido[i - 1]);  
    }  
    // Feldolgozás (algoritmus, stuki)  
    melyik = 2;  
    while (melyik <= n && !(mido[melyik - 1] < mido[melyik - 1 - 1])) {  
        melyik = melyik + 1;  
    }  
    van = melyik <= n;  
    // Kiírás (specifikáció ki)
```

melyik:=2

melyik ≤ n és
nem midő[melyik] < midő[melyik-1]

melyik:=melyik+1

van:=melyik ≤ n

}

Feladatmegoldási minta

gyorsabb vonat az előzőnél

Kiírás

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be, ki)  
    int n;  
    int[] mido;  
    bool van;  
    int melyik;  
    // Beolvasás (specifikáció be)  
    Console.Write("n = ");  
    int.TryParse(Console.ReadLine(), out n);  
    mido = new int[n];  
    for (int i = 1; i <= n; i++) {  
        Console.Write("{0}. menetido = ", i);  
        int.TryParse(Console.ReadLine(), out mido[i - 1]);  
    }  
    // Feldolgozás (algoritmus, stuki)  
    melyik = 2;  
    while (melyik <= n && !(mido[melyik - 1] < mido[melyik - 1 - 1])) {  
        melyik = melyik + 1;  
    }  
    van = melyik <= n;  
    // Kiírás (specifikáció ki)  
    if (van) {  
        Console.WriteLine("Van, a(z) {0}. vonat gyorsabb az előzőnél.", melyik);  
    }  
    else {  
        Console.WriteLine("Nincs gyorsabb vonat az előzőnél.");  
    }  
}
```

n = 6

1. menetido = 118

2. menetido = 200

3. menetido = 199

4. menetido = 116

5. menetido = 200

6. menetido = 122

Van, a(z) 3. vonat gyorsabb az előzőnél.

Ki: van ∈ L, melyik ∈ N

Feladatmegoldási minta gyorsabb vonat az előzőnél

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be,ki)  
    int n;  
    int[] mido;  
    bool van;  
    int melyik;  
    // Beolvasás (specifikáció be)  
    Console.Write("n = ");  
    int.TryParse(Console.ReadLine(), out n);  
    mido = new int[n];  
    for (int i = 1; i <= n; i++) {  
        Console.Write("{0}. menetido = ", i);  
        int.TryParse(Console.ReadLine(), out mido[i - 1]);  
    }  
    // Feldolgozás (algoritmus, stuki)  
    melyik = 2;  
    while (melyik <= n && !(mido[melyik - 1] < mido[melyik - 1 - 1])) {  
        melyik = melyik + 1;  
    }  
    van = melyik <= n;  
    // Kiírás (specifikáció ki)  
    if (van) {  
        Console.WriteLine(melyik);  
    }  
    else {  
        Console.WriteLine(-1);  
    }  
}
```

Kiírás módosítás

```
n = 6  
1. menetido = 118  
2. menetido = 200  
3. menetido = 199  
4. menetido = 116  
5. menetido = 200  
6. menetido = 122  
3
```

Kimenet

A standard kimenet első sorába egy az előzőnél gyorsabb vonat sorszámát kell írni (ha több ilyen is van, akkor az első)! Ha nincs ilyen vonat, akkor -1-et kell írni!

Feladatmegoldási minta gyorsabb vonat az előzőnél

Letöltési oldal

Dokumentum: Minta bemenet ▾

letölt

1

> gyozke > source > repos > elozonel-gyorsabb-vonat > bin > Debug > net6.0

Név

elozonel-gyorsabb-vonat.dll
elozonel-gyorsabb-vonat.exe
elozonel-gyorsabb-vonat.pdb
elozonel-gyorsabb-vonat.runtimeconfig.json
elozonel-gyorsabb-vonat.deps.json
ki1.txt
ki2.txt
be2.txt
be1.txt

Nemcsak a letöltött, de saját
tesztjeinket tehetjük fájlokba,
így sokkal gyorsabban
ellenőrizhetjük megoldásunk
helyességét.

2023. 10. 07. 15:02

2014. 09. 12. 20:55

2014. 09. 12. 20:55

2014. 09. 12. 20:53

2009. 12. 07. 14:41

3

be1.txt

```
4
90
80
70
60
```

ki1.txt

```
2
```

```
c:\Users\gyozke\source\repos\elozonel-gyorsabb-  
vonat\bin\Debug\net6.0>elozonel-gyorsabb-vonat.exe <be1.txt >ki1.txt
```

ki1.txt (saját)

```
n = 1. menetido = 2. menetido = 3. menetido =  
4. menetido = 2
```

Feladatmegoldási minta gyorsabb vonat az előzőnél

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be, ki)  
    int n;  
    int[] mido;  
    bool van;  
    int melyik;  
    // Beolvasás (specifikáció be)  
    Console.Error.Write("n = ");  
    int.TryParse(Console.ReadLine(), out n);  
    mido = new int[n];  
    for (int i = 1; i <= n; i++) {  
        Console.Error.Write("{0}. menetido = ", i);  
        int.TryParse(Console.ReadLine(), out mido[i - 1]);  
    }  
    // Feldolgozás (algoritmus, stuki)  
    melyik = 2;  
    while (melyik <= n && !(mido[melyik - 1] < mido[melyik - 1 - 1])) {  
        melyik = melyik + 1;  
    }  
}
```

```
n = 6  
1. menetido = 118  
2. menetido = 200  
3. menetido = 199  
4. menetido = 116  
5. menetido = 200  
6. menetido = 122  
3
```

```
c:\Users\gyozke\source\repos\elozonel-gyorsabb-  
vonat\bin\Debug\net6.0>elozonel-gyorsabb-  
vonat.exe <be1.txt >ki1.txt
```

```
if (van) {  
    Console.WriteLine(  
    }  
else {  
    Console.WriteLine(-1);  
    }  
}
```

ki1.txt

2



ki1.txt (saját)

2

Feladatmegoldási minta gyorsabb vonat az előzőnél

```
using System;
```

```
static void Main(string[] args) {  
    // Deklarálás (változók, specifikáció be,ki)  
    int n;  
    int[] mido;  
    bool van;
```

Téma

Feladat

Mintafeladat

Megoldom

Eredmény

Letölt

Feladat beadása

Feladat: Gyorsabb vonat az előzőnél *

Feladat nyelv:

☐ C

☐ C++

☒ C#

19 próbálkozás maradt

Fájl kiválasztása Nincs fájl kiválasztva

beadom

Utolsó beadás eredménye

Összpont: 100/100

Teszt#	Pont	Üzenet...	Futási idő
1.1	3/3	Helyes	0.032 sec
2.1	3/3	Helyes	0.032 sec
3.1	3/3	Helyes	0.035 sec
4.1	3/3	Helyes	0.031 sec
5.1	3/3	Helyes	0.032 sec
6.1	3/3	Helyes	0.033 sec
7.1	3/3	Helyes	0.033 sec
8.1	3/3	Helyes	0.032 sec
9.1	4/4	Helyes	0.037 sec
10.1	4/4	Helyes	0.033 sec
11.1	4/4	Helyes	0.032 sec
12.1	4/4	Helyes	0.036 sec
13.1	4/4	Helyes	0.037 sec

```
while (melyik <= n && !(mido[melyik - 1] < mido[melyik - 1 - 1])  
    melyik = melyik + 1;  
}  
van = melyik <= n;  
// Kiírás (specifikáció ki)  
if (van) {  
    Console.WriteLine(melyik);  
}  
else {  
    Console.WriteLine(-1);  
}
```

Feladatmegoldási minta



Feladatmegoldási minta

legmagasabb tanuló

Ismerjük egy osztály tanulóinak magasságait.
Mondd meg, hányadik diák a legmagasabb, és a magasságát is!

mag		
m cm		
1	1	52
2	1	77
3	1	23
4	1	65



hol =2,
maxmag=(m:1, cm:77)

Feladatmegoldási minta

legmagasabb tanuló

Feladat:

maximumkiválasztás

Ismerjük egy osztály tanulóinak magasságait. Mondd meg, hányadik diák a **leg**magasabb, és a magasságát is!

Specifikáció:

Be: $\text{mag} \in \text{Mag}[]$, $\text{Mag} = (\text{m}:\mathbb{N} \times \text{cm}:\mathbb{N})$

Ki: $\text{hol} \in \mathbb{N}$, $\text{maxmag} \in \text{Mag}$

Ef: $\text{hossz}(\text{mag}) \geq 1$

Uf: $\text{hol} \in [1..\text{hossz}(\text{mag})]$ és

$\forall i \in [1..\text{hossz}(\text{mag})]:$

$\text{mag}[\text{hol}].\text{m} > \text{mag}[i].\text{m}$ vagy

$(\text{mag}[\text{hol}].\text{m} = \text{mag}[i].\text{m} \text{ és } \text{mag}[\text{hol}].\text{cm} \geq \text{mag}[i].\text{cm}))$ és

$\text{maxmag} = \text{mag}[\text{hol}]$

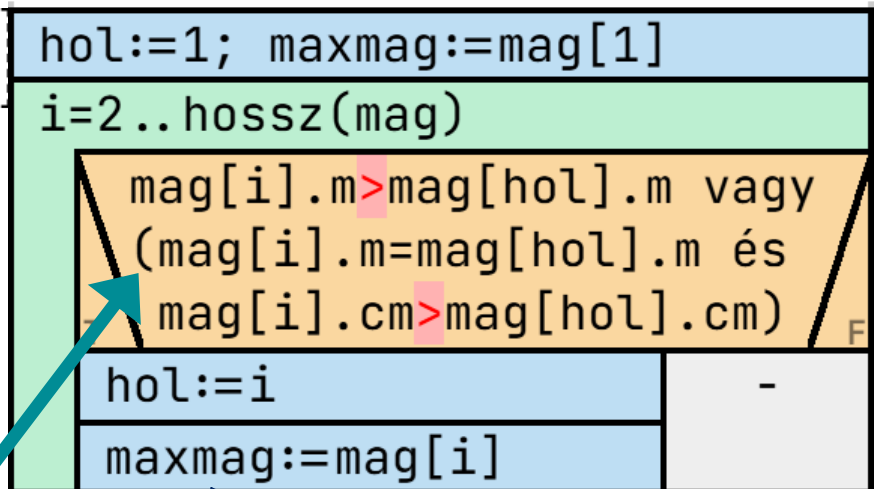
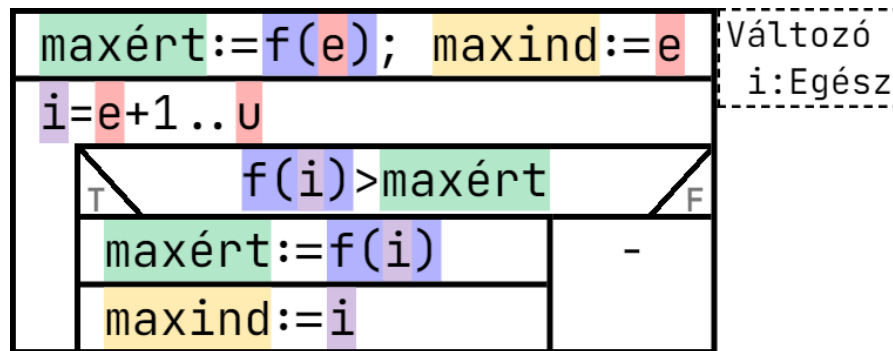
mag

m cm

1	1	52
2	1	77
3	1	23
4	1	65

Feladatmegoldási minta legmagasabb tanuló

Algoritmus **analóg algoritmikus gondolkodással:**



Be: mag ∈ Mag[], Mag=(m:N x cm:N)

Ki: hol ∈ N, maxmag ∈ Mag

Ef: hossz(mag) ≥ 1

Uf: hol ∈ [1..hossz(mag)] és

∀ i ∈ [1..hossz(mag)]:

mag[hol].m > mag[i].m vagy
(mag[hol].m = mag[i].m és mag[hol].cm ≥ mag[i].cm) és
maxmag = mag[hol]

Ez nem visszavezetés!

Feladatmegoldási minta

legmagasabb tanuló

Algoritmus analóg algoritmikus gondolkodással:

maxért:=f(e); maxind:=e		Változó i:Egész
i=e+1..u		
T	f(i)>maxért	F
maxért:=f(i)		-
maxind:=i		

Változó i:Egész

hol:=1; maxmag:=mag[1]	
i=2..hossz(mag)	
<div>mag[i].m>mag[hol].m vagy (mag[i].m=mag[hol].m és mag[i].cm>mag[hol].cm)</div>	
hol:=i	-
maxmag:=mag[i]	

Local declarations

Változó i:Egész

hol:=1; maxmag:=mag[1]		
i=2..hossz(mag)		
T	mag[i].m>mag[hol].m	
hol:=i		mag[i].m=mag[hol].m és mag[i].cm>mag[hol].cm
maxmag:=mag[i]		
hol:=i		-
maxmag:=mag[i]		



Feladatmegoldási minta

legmagasabb tanuló

Feladat:

maximumkiválasztás

Ismerjük egy osztály tanulóinak magasságait. Mondd meg, hányadik diák a legmagasabb, és a magasságát is!

Specifikáció:

Be: $\text{mag} \in \text{Mag}[]$, $\text{Mag} = (m:N \times \text{cm}:N)$

Ki: $ho \in N$, $\text{maxmag} \in \text{Mag}$

Ef: $\text{hossz}(\text{mag}) \geq 1$

Uf: $(\text{maxind}, \text{maxért}) = \text{MAX}(i=e..u, f(i))$

i	f(i)
e	f(e)
e+1	f(e+1)
e+2	f(e+2)
...	...
u-2	f(u-2)
u-1	f(u-1)
u	f(u)

maxind, maxért

	mag	
	m	cm
1	1	52
2	1	77
3	1	23
4	1	65

1. Mik az intervallum határai?
2. Milyen értékeket veszünk az intervallum egyes pontján, amik közül a legnagyobb kikerül?
3. Milyen néven tároljuk az eredményeket?

A helyes $f(i)$ megtalálása a cél!
Hogyan hasonlíthatunk össze rekordokat?
Hogyan rendelhetünk egyetlen értéket a rekordhoz, amiket összehasonlíthatunk?

Feladatmegoldási minta

legmagasabb tanuló

Feladat:

maximumkiválasztás

Ismerjük egy osztály tanulóinak magasságait. Mondd meg, hányadik diák a legmagasabb, és a magasságát is!

Specifikáció:

Be: $\text{mag} \in \text{Mag}[]$, $\text{Mag} = (m:N \times \text{cm}:N)$

Sa: $\text{maxcm} \in N$

Ki: $\text{hol} \in N$, $\text{maxmag} \in \text{Mag}$

Ef: $\text{hossz}(\text{mag}) \geq 1$

Uf: $(\text{hol}, \text{maxcm}) = \text{MAX}(i=1.. \text{hossz}(\text{mag}),$
 $\text{mag}[i].m * 100 + \text{mag}[i].\text{cm})$ és
 $\text{maxmag} = \text{mag}[\text{hol}]$

i	f(i)
e	f(e)
e+1	f(e+1)
e+2	f(e+2)
...	...
u-2	f(u-2)
u-1	f(u-1)
u	f(u)

maxind, maxért

mag				
	m	cm		
1	1	52		
2	1	77		
3	1	23		
4	1	65		

m*100+cm	
	152
	177
	123
	165

Az $f(i)$ a rekord adatkettesét képezi le egyetlen értékre, és ez kerül összehasonlításra.

Feladatmegoldási minta

legmagasabb tanuló

Feladatsablon

Be: $e \in \mathbb{Z}, u \in \mathbb{Z}$

Ki: $\text{maxind} \in \mathbb{Z}, \text{maxért} \in \mathbb{H}$

Ef: $e \leq u$

Uf: $(\text{maxind}, \text{maxért}) =$

$\text{MAX}(i = e..u, f(i))$

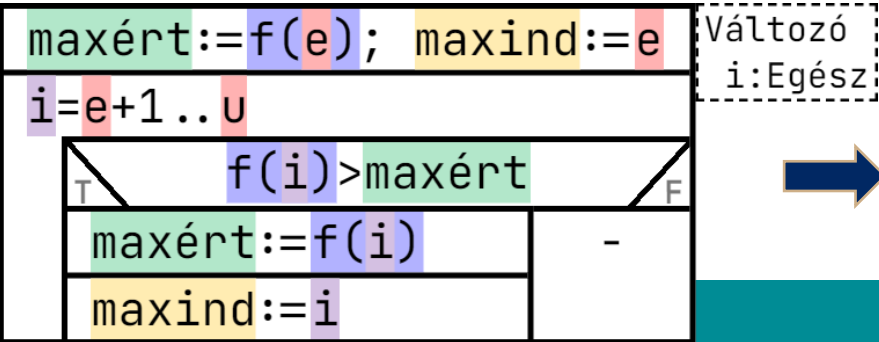
Visszavezetés:

$\text{maxind}, \text{maxért} \sim \text{hol}, \text{maxcm}$

$e..u \sim 1..hossz(mag)$

$f(i) \sim mag[i].m * 100 + mag[i].cm$

Algoritmus:



Legmagasabb tanuló

Be: $mag \in \text{Mag}[], \text{Mag} = (m:N \times cm:N)$

Sa: $\text{maxcm} \in \mathbb{N}$

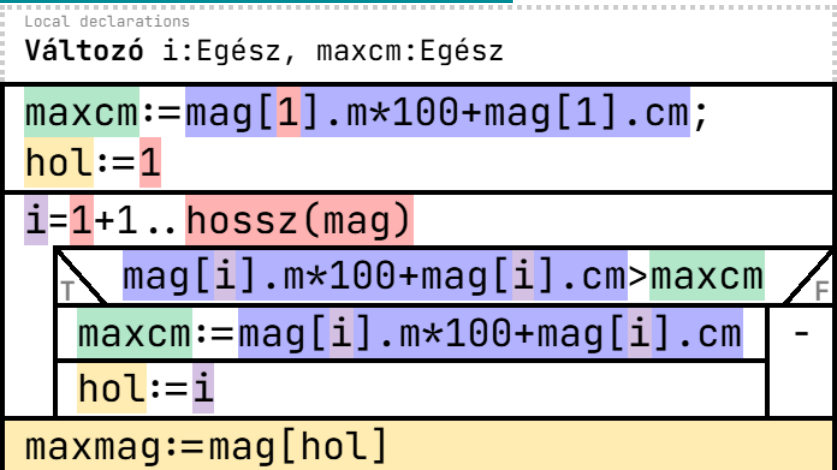
Ki: $\text{hol} \in \mathbb{N}, \text{maxmag} \in \text{Mag}$

Ef: $\text{hossz}(mag) \geq 1$

Uf: $(\text{hol}, \text{maxcm}) = \text{MAX}(i = 1..hossz(mag),$

$mag[i].m * 100 + mag[i].cm)$ és

$\text{maxmag} = mag[\text{hol}]$



Feladatmegoldási minta

legmagasabb tanuló

Feladat:

maximumkiválasztás

Ismerjük egy osztály tanulóinak magasságait. Mondd meg, hányadik diák a legmagasabb, és a magasságát is!

Specifikáció:

Be: $\text{mag} \in \text{Mag}[]$, $\text{Mag} = (m:N \times \text{cm}:N)$

Sa: $\text{maxcm} \in N$

Ki: $\text{hol} \in N$, $\text{maxmag} \in \text{Mag}$

$f(i)$

Fv: $f:N \rightarrow N$, $f(i) = \text{mag}[i].m * 100 + \text{mag}[i].\text{cm}$

Ef: $\text{hossz}(\text{mag}) \geq 1$

Uf: $(\text{hol}, \text{maxcm}) = \text{MAX}(i=1..\text{hossz}(\text{mag}), f(i))$ és
 $\text{maxmag} = \text{mag}[\text{hol}]$

i	f(i)
e	f(e)
e+1	f(e+1)
e+2	f(e+2)
...	...
u-2	f(u-2)
u-1	f(u-1)
u	f(u)

maxind, maxért

mag			
	m	cm	m*100+cm
1	1	52	152
2	1	77	177
3	1	23	123
4	1	65	165



Feladatmegoldási minta



Feladatmegoldási minta

lájknövekedések száma

A Youtube minden napra megmondja egy videónkról, hogy hány lájkot kapott. Hány nap növekedett a lájkok száma az előző naphoz képest? (Az első nap előtt 0 lájkunk volt.)

lájk	
1	5
2	5
3	15
4	25

→ db=3

lájk	
1	0
2	6
3	15
4	25

→ db=3

Feladatmegoldási minta

lájknövekedések száma

Feladat:

megszámolás

Hány nap növekedett a lájkok száma az előző naphoz képest?

Specifikáció:

Be: lájkokszáma $\in\mathbb{N}[]$

Ki: hány $\in\mathbb{N}$

Ef: -

Uf: db=DARAB($i=e..u$, $T(i)$)

1. Mik az intervallum határai?
2. Milyen értékeket veszünk az intervallum egyes pontján, amik közül a legnagyobb kikerül?
3. Milyen néven tároljuk az eredményeket?

A helyes $T(i)$ megtalálása a cél!

i	T(i)	érték
e	IGAZ	1
e+1	HAMIS	0
e+2	HAMIS	0
...
u-2	IGAZ	1
u-1	IGAZ	1
u	HAMIS	0
		=
		db

lájknövekedések száma

1	5
2	5
3	15
4	25

hány=3

Feladatmegoldási minta

lájknövekedések száma

Feladat:

megszámolás

Hány nap növekedett a lájkok száma az előző naphoz képest?

Specifikáció:

Be: lájkokszáma $\in\mathbb{N}[]$

Ki: hány $\in\mathbb{N}$

Ef: -

Uf: hány=**DARAB**($i=1..hossz(lájkokszáma)$,

($i=1 \rightarrow lájkokszáma[i]>0$) és

($i>1 \rightarrow lájkokszáma[i]>lájkokszáma[i-1]$))

T(i)

i	T(i)	érték
e	→ IGAZ	1
e+1	→ HAMIS	0
e+2	→ HAMIS	0
...	→
u-2	→ IGAZ	1
u-1	→ IGAZ	1
u	→ HAMIS	0
		=
		db

lájkk

1	5
2	5
3	15
4	25

→ hány=3

Feladatmegoldási minta

lájknövekedések száma

Feladatsablon

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $db \in \mathbb{N}$

Ef: -

Uf: $db = \text{DARAB}(i=e..u, T(i))$

Lájknövekedések száma

Be: $\text{lájkokszáma} \in \mathbb{N}[]$

Ki: $\text{hány} \in \mathbb{N}$

Ef: -

Uf: $\text{hány} = \text{DARAB}(i=1..\text{hossz}(\text{lájkokszáma}),$
 $(i=1 \rightarrow \text{lájkokszáma}[i] > 0) \text{ és}$
 $(i > 1 \rightarrow$
 $\text{lájkokszáma}[i] > \text{lájkokszáma}[i-1]))$



Visszavezetés:

$db \sim \text{hány}$

$e..u \sim 1..\text{hossz}(\text{lájkokszáma})$

$T(i) \sim \begin{cases} (i=1 \rightarrow \text{lájkokszáma}[i] > 0) \text{ és} \\ (i > 1 \rightarrow \text{lájkokszáma}[i] > \text{lájkokszáma}[i-1]) \end{cases}$

Feladatmegoldási minta

lájknövekedések száma

Feladatsablon

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $db \in \mathbb{N}$

Ef: -

Uf: $db = \text{DARAB}(i = e..u, T(i))$

Lájknövekedések száma

Be: $\text{lájkokszáma} \in \mathbb{N}[]$

Ki: $\text{hány} \in \mathbb{N}$

Ef: -

Uf: $\text{hány} = \text{DARAB}(i = 1..hossz(\text{lájkokszáma}),$
 $(i = 1 \text{ és } \text{lájkokszáma}[i] > 0) \text{ vagy}$
 $(i > 1 \text{ és}$
 $\text{lájkokszáma}[i] > \text{lájkokszáma}[i-1]))$



Visszavezetés:

$db \sim \text{hány}$

$e..u \sim 1..hossz(\text{lájkokszáma})$

$T(i) \sim \begin{cases} (i = 1 \text{ és } \text{lájkokszáma}[i] > 0) \text{ vagy} \\ (i > 1 \text{ és } \text{lájkokszáma}[i] > \text{lájkokszáma}[i-1]) \end{cases}$

Feladatmegoldási minta

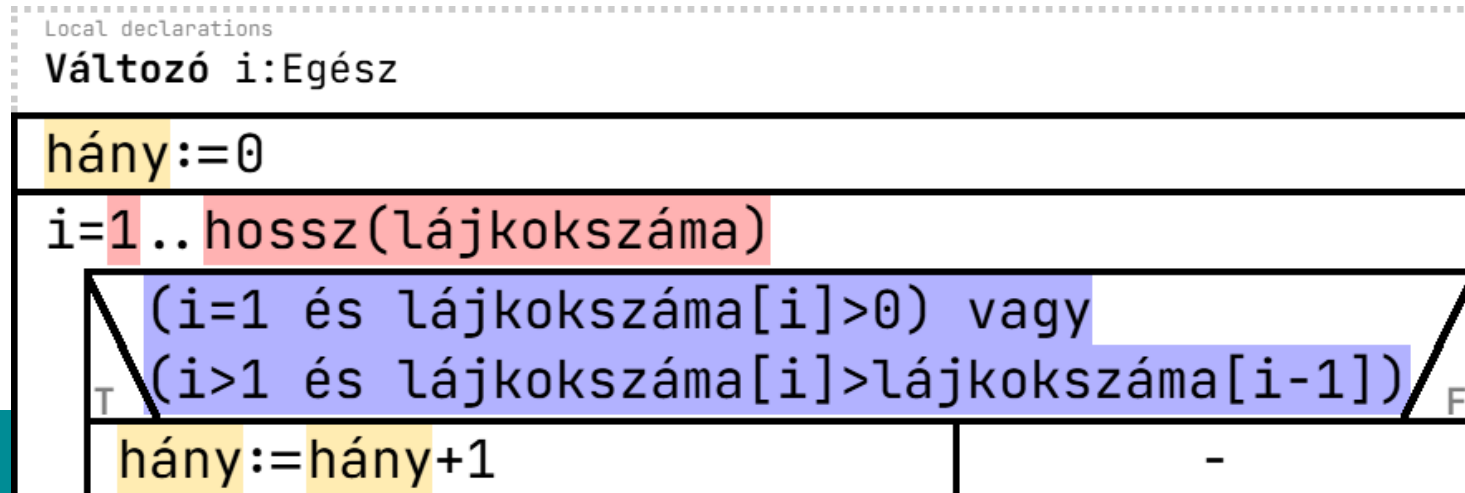
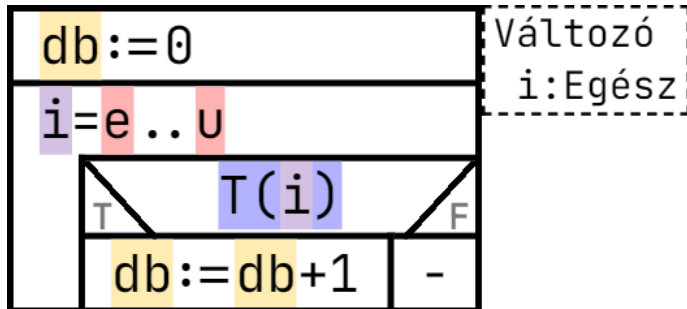
lájknövekedések száma

Algoritmus

db ~ hány

e..u ~ 1..hossz(lájkokszáma)

$T(i) \sim \begin{cases} i=1 \text{ és } \text{lájkokszáma}[i]>0 & \text{vagy} \\ i>1 \text{ és } \text{lájkokszáma}[i]>\text{lájkokszáma}[i-1] \end{cases}$



Feladatmegoldási minta

lájknövekedések száma – másképp

Feladat:

Hány nap növekedett a lájkok száma az előző naphoz képest?

Specifikáció:

Be: lájkokszáma $\in\mathbb{N}[]$

Sa: db $\in\mathbb{N}$

Ki: hány $\in\mathbb{N}$

Ef: -

Uf: db=**DARAB**(i=2.. $\text{hossz}(\text{lájkokszáma})$,
lájkokszáma[i]>lájkokszáma[i-1]) és
lájkokszáma[1]>0 -> hány=db+1 és
nem(lájkokszáma[1]>0) -> hány=db

i	T(i)	érték
e	→ IGAZ	1
e+1	→ HAMIS	0
e+2	→ HAMIS	0
...	→
u-2	→ IGAZ	1
u-1	→ IGAZ	1
u	→ HAMIS	0
		=
		db

lájkk

1	5
2	5
3	15
4	25

→ hány=3

Az esetszétválasztás a mintán kívül van

Feladatmegoldási minta

lájknövekedések száma – másképp

Feladatsablon

Be: $e \in \mathbb{Z}$, $u \in \mathbb{Z}$

Ki: $db \in \mathbb{N}$

Ef: -

Uf: $db = \text{DARAB}(i=e..u, T(i))$

Lájknövekedések száma

Be: $\text{lájkokszáma} \in \mathbb{N}[]$

Sa: $db \in \mathbb{N}$

Ki: $\text{hány} \in \mathbb{N}$

Ef: -

Uf: $db = \text{DARAB}(i=2..\text{hossz}(\text{lájkokszáma}), \text{lájkokszáma}[i] > \text{lájkokszáma}[i-1])$

és $\text{lájkokszáma}[1] > 0 \rightarrow \text{hány} = db + 1$ és
 $\text{nem}(\text{lájkokszáma}[1] > 0) \rightarrow \text{hány} = db$

Visszavezetés:

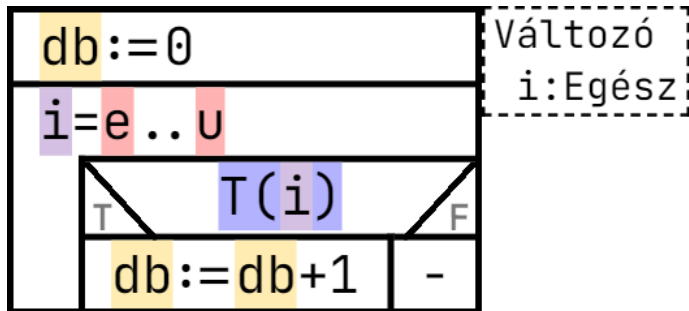
$e..u \sim 1..\text{hossz}(\text{lájkokszáma})$

$T(i) \sim \text{lájkokszáma}[i] > \text{lájkokszáma}[i-1]$

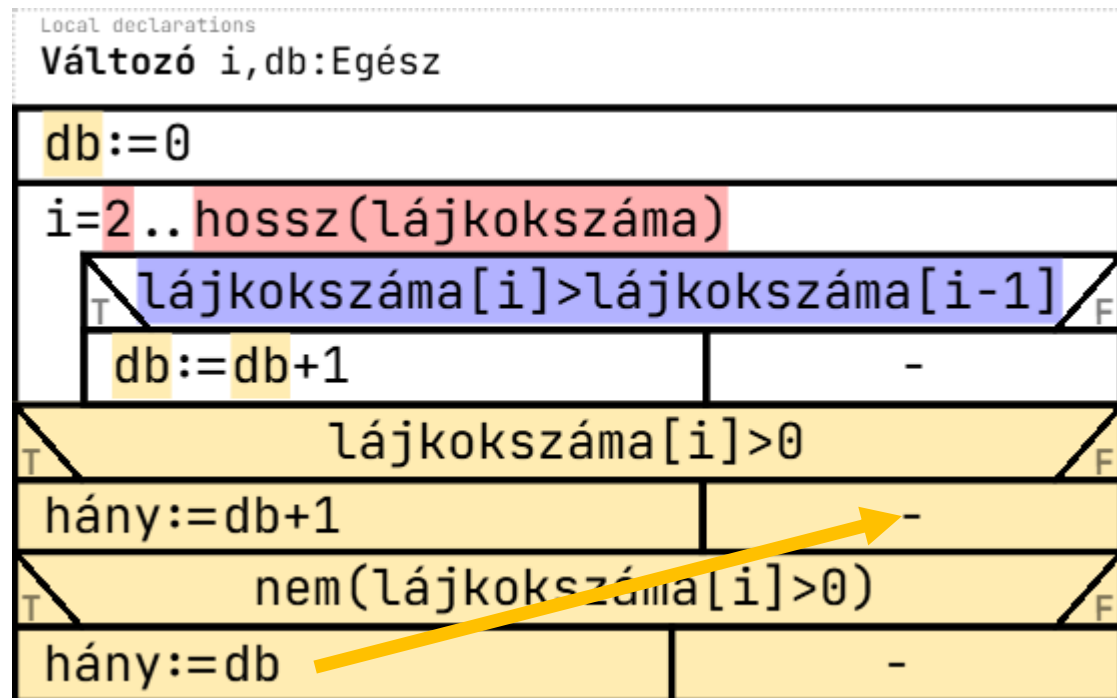
Feladatmegoldási minta

lájknövekedések száma – másképp

Algoritmus



e..u ~ 1..hossz(lájkokszáma)
T(i) ~ lájkokszáma[i]>lájkokszáma[i-1]

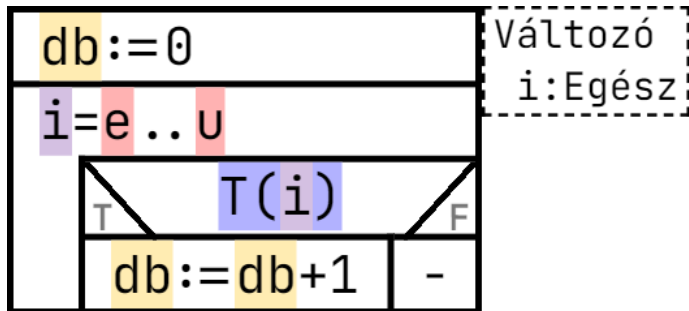


uf: db=DARAB(i=2..hossz(lájkokszáma),
lájkokszáma[i]>lájkokszáma[i-1]) és
lájkokszáma[1]>0 -> hány=db+1 és
nem(lájkokszáma[1]>0) -> hány=db

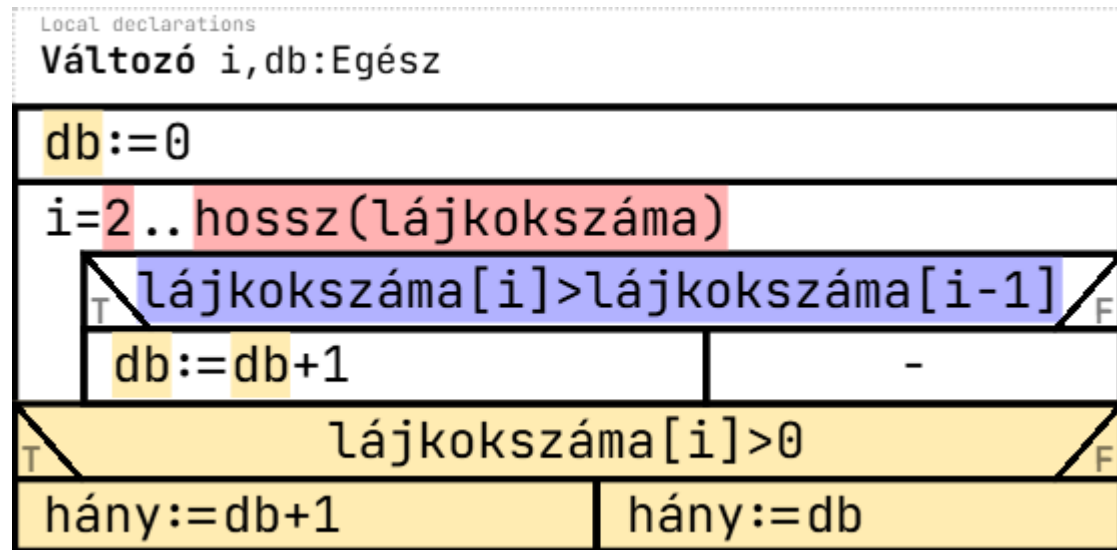
Feladatmegoldási minta

lájknövekedések száma – másképp

Algoritmus



$e..u \sim 1..hossz(lájkokszáma)$
 $T(i) \sim lájkokszáma[i]>lájkokszáma[i-1]$



uf: db=DARAB(i=2..hossz(lájkokszáma),
 lájkokszáma[i]>lájkokszáma[i-1]) és
 lájkokszáma[1]>0 -> hány=db+1 és
 nem(lájkokszáma[1]>0) -> hány=db