

# PROGRAMOZÁS Mintamegoldás

Horváth Győző, Pluhár Zsuzsa



# Ismétlés



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





#### Összegzés

### i f(i) $e \rightarrow f(e)$ $e+1 \rightarrow f(e+1)$ $e+2 \rightarrow f(e+2)$ ... $\rightarrow$ ... $u-2 \rightarrow f(u-2)$ $u-1 \rightarrow f(u-1)$ $u \rightarrow f(u)$ = 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-1 → HAMIS 0
=
db
```

# Maximum kiválasztás

```
i f(i)

e \rightarrow f(e)

e+1 \rightarrow f(e+1)

e+2 \rightarrow f(e+2)

... \rightarrow ...

u-2 \rightarrow f(u-2)

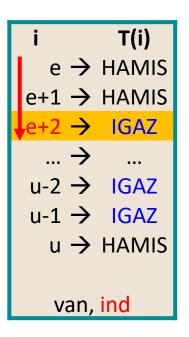
u-1 \rightarrow f(u-1)

u \rightarrow f(u)
```

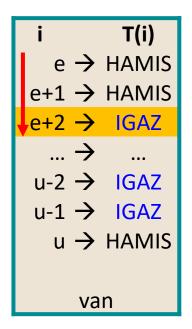
#### Feltételes maximumkeresés

```
i T(i) f(i)
e \rightarrow HAMIS \quad f(e)
e+1 \rightarrow IGAZ \quad f(e+1)
e+2 \rightarrow IGAZ \quad f(e+2)
... \rightarrow ...
u-2 \rightarrow HAMIS \quad f(u-2)
u-1 \rightarrow IGAZ \quad f(u-1)
van, maxind, maxért
```

#### Keresés



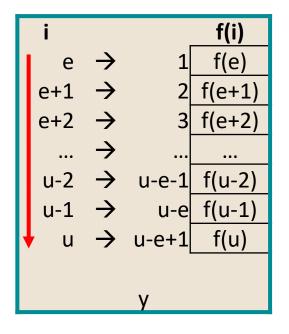
#### Eldöntés



#### Kiválasztás

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

#### Másolás



#### 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)

db, y
```

# Feladatmegoldási minta



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≤N≤100). A következő N sor mindegyike egy-egy egész számot tartalmaz, az egyes vonatok menetidejét (1≤M≤1000).

#### Kimenet

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

#### Példa

Bemenet	Kimenet
6	3
118	
200	
199	
116	
200	
122	

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:  $n \in \mathbb{N}$ ,  $mid \in \mathbb{N}[1..n]$ 

Ki: van∈L, melyik∈N

Ef: -

Uf: (van, melyik) = KERES(i=2..n, midő[i] < midő[i-1])</pre>

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

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

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

# Feladatsablon (mintafeladat) (konkrét feladat) Be: $e \in Z$ , $u \in Z$ Be: $n \in N$ , $midő \in N[1..n]$ Ki: $van \in L$ , $ind \in Z$ Ki: $van \in L$ , $melyik \in N$ Ef: Uf: (van,ind) = KERES(i=e..u, Uf: (van,melyik) = KERES(i=2..n, T(i))ind ~ melyike..u ~ 2..n

 $T(i) \sim mid\sigma[i] < mid\sigma[i-1]$ 

```
ind:=e
ind ≤ u és nem T(ind)
ind:=ind+1
van:=ind ≤ u

FITELIK
```

```
melyik:=2
melyik≤n és
nem midő[melyik]<midő[melyik-1]
    melyik:=melyik+1
van:=melyik≤n</pre>
```

```
static void Main(string[] args) {
 // Deklarálás (változók, specifikáció be,ki)
 // Beolvasás (specifikáció be)
 // Feldolgozás (algoritmus, stuki)
 // Kiírás (specifikáció ki)
```

Kódolás alapsablonja

```
static void Main(string[] args) {
 // Deklarálás (változók, specifikáció be,ki)
 fint n;
                           Be: n \in \mathbb{N}, midő\in \mathbb{N}[1..n]
 int[] mido;
 bool van;
                           Ki: van∈L, melyik∈N
 int melyik;
 // Beolvasás (specifikáció be)
 // Feldolgozás (algoritmus, stuki)
 // Kiírás (specifikáció ki)
```

#### Deklarálás

```
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 = ");
                                                    Be: n∈N,
 int.TryParse(Console.ReadLine(), out n);
 fmido = new int[n];
 for (int i = 1; i <= n; i++) {
                                                         midő∈N[1..n]
   Console.Write("{0}. menetido = ", i);
   int.TryParse(Console.ReadLine(), out mido[i - 1]);
 // Feldolgozás (algoritmus, stuki)
 // Kiírás (specifikáció ki)
```

```
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 = 0; i < n; i++) {
   Console.Write("{0}. menetido = ", i);
   int.TryParse(Console.ReadLine(), out mido[i]);
 // Feldolgozás (algoritmus, stuki)
 melyik = 1;
 while (melyik < n && !(mido[melyik] < mido[melyik - 1])) {</pre>
   melyik = melyik + 1;
                                           melyik:=2
 van = melyik < n;
                                           melyik≤n és
 // Kiírás (specifikáció ki)
                                           nem midő[melyik]<midő[melyik-1]</pre>
                                               melyik:=melyik+1
                                            van:=melyik≤n
```

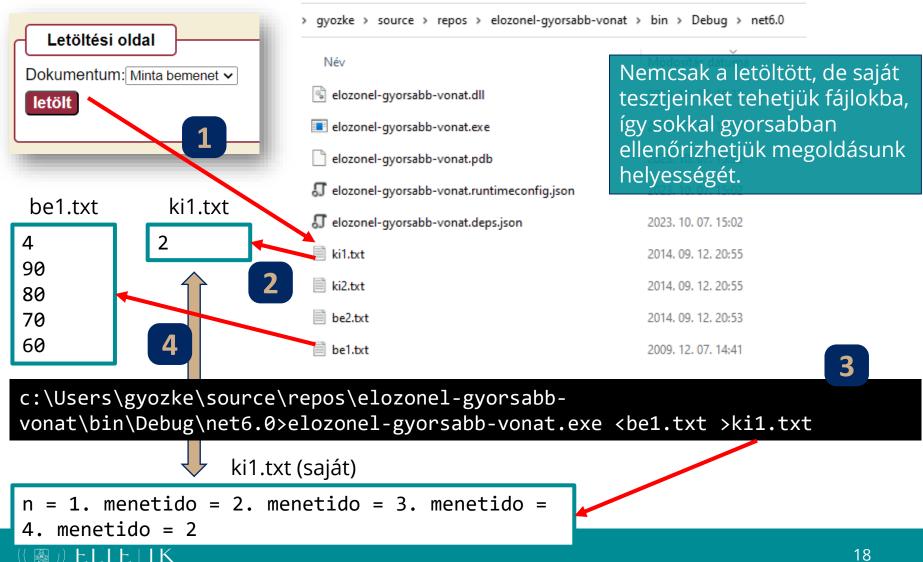
```
Kiírás
static void Main(string[] args) {
 // Deklarálás (változók, specifikáció be,ki
                                             n = 6
 int n;
                                             1. menetido = 118
 int[] mido;
 bool van;
                                             2. menetido = 200
 int melyik;
                                             3. menetido = 199
 // Beolvasás (specifikáció be)
                                             4. menetido = 116
 Console.Write("n = ");
                                             5. menetido = 200
 int.TryParse(Console.ReadLine(), out n);
 mido = new int[n];
                                             6. menetido = 122
 for (int i = 0; i < n; i++) {
                                             Van, a(z) 3. vonat gyorsabb az előzőnél.
   Console.Write("{0}. menetido = ", i);
   int.TryParse(Console.ReadLine(), out mido[i]);
 // Feldolgozás (algoritmus, stuki)
 melyik = 1;
 while (melyik < n && !(mido[melyik] < mido[melyik - 1])) {</pre>
   melyik = melyik + 1;
 van = melyik < n;</pre>
 // Kiírás (specifikáció ki)
                                                             Ki: van∈L, melyik∈N
 if (van) {
   Console.WriteLine("Van, a(z) {0}. vonat gyorsabb az előzőnél.", melyik+1);
  else {
   Console.WriteLine("Nincs gyorsabb vonat az előzőnél.");
```

```
static void Main(string[] args) {
 int n;
 int[] mido;
 bool van;
 int melyik;
 Console.Write("n = ");
 int.TryParse(Console.ReadLine(), out n);
 mido = new int[n];
 for (int i = 0; i < n; i++) {
   Console.Write("{0}. menetido = ", i);
    int.TryParse(Console.ReadLine(), out mido[i]);
 melyik = 1;
 while (melyik < n && !(mido[melyik] < mido[melyik - 1])) {</pre>
   melyik = melyik + 1;
 van = melyik < n;</pre>
 // Kiírás (specifikáció ki)
 if (van) {
   Console.WriteLine(melyik+1);
                               Kimenet
  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
```

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



```
static void Main(string[] args) {
                                                             n = 6
                                                              1. menetido = 118
 int n;
 int[] mido;
                                                              2. menetido = 200
 bool van;
                                                              3. menetido = 199
 int melyik;
                                                              4. menetido = 116
 // Beolvasás (specifikáció be)
                                                              5. menetido = 200
 Console.Error.Write("n = ");
 int.TryParse(Console.ReadLine(), out n);
                                                              6. menetido = 122
 mido = new int[n];
                                                              3
 for (int i = 0; i < n; i++) {
   Console.Error.Write("{0}. menetido = ", i);
   int.TryParse(Console.ReadLine(), out mido[i]);
 // Feldolgozás (algoritmus, stuki)
 melyik = 1;
 while (melyik < n && !(mido[melyik] < mido[melyik - 1])) {</pre>
c:\Users\gyozke\source\repos\elozonel-gyorsabb-
vonat\bin\Debug\net6.0>elozonel-gyorsabb-vonat.exe <be1.txt >ki1.txt
 // Kiiras (specitikacio ki)
 if (van) {
   Console.WriteLin
                     ki1.txt
                                                            ki1.txt (saját)
                                                          2
 else {
   Console.WriteLine(-1);
```

