

Algoritmusok és adatszerkezetek

Göncziné Kapros Katalin
gonczone.kapros.katalin@gde.hu
Kapros Károly
kapros.karoly@gde.hu

Gyak



Adatsorhoz egy adatot rendelő eljárások

Tömbfeltöltés tétele

- Deklaráltuk az n elemű a vektort. A vektor feltöltésének algoritmus:

```
eljárás TÖMBFELTÖLTÉS  
  ciklus I = 0-tól (N - 1)-ig  
    be: A[I]  
  ciklus vége  
eljárás vége
```

- Ezt az algoritmust akkor használhatjuk, ha a vektort inputról (pl. billentyűzetről) akarjuk feltölteni. A harmadik sor be: A[i] utasítása helyett írható pl. értékadás is, ha a vektor elemeit pl. egy másik tömb értékeinek felhasználásával akarjuk megadni.

Összegzés tétele 1

- Az összegzés tétele egy vektor elemeinek összegét (szorzatát, számtani közepét stb.) adja meg. Feltételezzük, hogy a vektor elemei numerikus típusúak.
- Feladat: Adott az N elemű A vektor. Keressük az elemei összegét (szorzatát, számtani közepét).

```
eljárás ÖSSZEGZÉS1  
  ÖSSZEG = 0  
  ciklus I = 0-tól (N - 1)-ig  
    ÖSSZEG = ÖSSZEG + A[I]  
  ciklus vége  
ki: ÖSSZEG  
eljárás vége
```

Összegzés példa

- Feladat: Tegyük fel, hogy bevásárlás után szeretnénk a pénztárblokk adatait ellenőrizni. A TÉTEL tömbben tároltuk a vásárolt termékek árait, és szeretnénk ellenőrzésként kiszámítani a végösszeget.

sorszám	TÉTEL
1	427
2	1271
3	57
4	983
5	234

I	TÉTEL[I]	ÖSSZEG
–	–	0
0	427	427
1	1271	1698
2	57	1755
3	983	2738
4	234	2972

eljárás ÖSSZEGZÉS1

ÖSSZEG = 0

ciklus I = 0-tól (N – 1)-ig

ÖSSZEG = ÖSSZEG + A[I]

ciklus vége

ki: ÖSSZEG

eljárás vége

Összegzés tétele 2

- Abban az esetben, ha egy vektor összes elemének összegét keressük, az ÖSSZEG változó kezdőértékének beállíthatjuk a tételek közül az elsőt, és a ciklust a második elemtől indíthatjuk.

- Hogyan lesz az összegből átlag?
Amennyiben $N > 0$

```
ki: ÖSSZEG / N
```

- Ha az elemek szorzatát keressük?

```
SZORZAT = SZORZAT * A[I]
```

Mi legyen a szorzat változó kezdő értéke? Lehet nulla?

```
eljárás ÖSSZEGZES2  
  ÖSSZEG = A[0]  
  ciklus I = 1-től (N - 1)-ig  
    ÖSSZEG = ÖSSZEG + A[I]  
  ciklus vége  
ki: ÖSSZEG  
eljárás vége
```


Összegzés tétele 3

- Az összegzés algoritmusát alkalmazzuk akkor is, ha egy vektornak csak bizonyos elemeit szeretnénk összegezni, ekkor viszont feltételes összegzést alkalmazunk.
- Feladat: a TÉTELEK vektorban tároljuk egy háztartás bevételeit és kiadásait. A bevételek pozitív, a kiadások negatív előjelűek. Mennyi a háztartás teljes bevétele, teljes kiadása és jelenlegi egyenlege?

eljárás PÉLDA1

BEVÉTEL = 0

KIADÁS = 0

ciklus I = 0-tól (N - 1)-ig

 ha TÉTEL[I] > 0 akkor

 BEVÉTEL = BEVÉTEL + TÉTEL[I]

 különben

 KIADÁS = KIADÁS + TÉTEL[I]

 elágazás vége

ciklus vége

ki: BEVÉTEL, KIADÁS, BEVÉTEL + KIADÁS

eljárás vége

Eldöntés tétele

- Adott egy N elemű A vektor és egy T tulajdonság, amely értelmezhető a tömb elemeire (páros, nemnegatív, osztható öttel, hosszabb tíz karakternél, van benne 'k' betű stb.).
- Kérdés: van-e az A vektornak T tulajdonságú eleme?
- Elegendő csak addig keresnünk T tulajdonságú elemet míg a legelsőt meg nem találjuk
Legkedvezőtlenebb eset: nincs a tömbben ilyen elem (N , vagyis tömbelem indexe: $N-1$)

Eldöntés tétele

Matematikai azonosságok:

$$\neg(P \wedge Q) = \neg P \vee \neg Q$$

$$\neg(P \vee Q) = \neg P \wedge \neg Q$$

A SZÁMOK vektorban természetes számokat tárolunk.

Kérdés: van-e közöttük olyan szám, amelyik 5-nél nagyobb, de 10-nél kisebb?

T: SZÁMOK[I] > 5 és SZÁMOK[I] < 10

$\neg$ T: SZÁMOK[I] ≤ 5 vagy SZÁMOK[I] ≥ 10

eljárás eldöntés

I = 0

ciklus, amíg (I < N) és (A[I] nem T tulajdonságú)

I = I + 1

ciklus vége

ha I < N, akkor

ki: "Igen, van T tulajdonságú elem"

különben

ki: "Nem, nincs T tulajdonságú elem"

elágazás vége

eljárás vége

eljárás példa2

I = 0

ciklus, amíg (I < N) és (SZÁMOK[I] ≤ 5 vagy SZÁMOK[I] ≥ 10)

I = I + 1

ciklus vége

ha I < N akkor

ki: "Igen, van olyan szám közöttük, amelyik 5-10 közé esik"

különben

ki: "Nem, nincs ilyen szám."

elágazás vége

eljárás vége

Megszámlálás tétele

- Adott az N elemű A vektor és egy T tulajdonság. Kérdés: hány T tulajdonságú eleme van A -nak?

```
eljárás megszámolás
```

```
    DB = 0
```

```
    ciklus I = 0-tól (N - 1)-ig
```

```
        ha A[I] T tulajdonságú, akkor
```

```
            DB = DB + 1
```

```
        elágazás vége
```

```
    ciklus vége
```

```
    ki: DB
```

```
eljárás vége
```

Megszámlálás példa

- Ismerjük egy 15 fős iskolai csoport tanulójának nemét és magasságát cm-ben. Számítsuk ki a fiúk átlagos magasságát!

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
NEM	F	F	N	N	N	F	F	N	F	N	F	N	N	F	F
MAGASSÁG	178	192	167	159	164	169	182	177	165	170	170	173	164	181	172

```
eljárás példa3
  ÖSSZEG = 0
  DB = 0
  ciklus I = 0-tól 14-ig
    ha NEM[I] = "F", akkor
      DB = DB + 1
      ÖSSZEG = ÖSSZEG + MAGASSÁG[I]
    elágazás vége
  ciklus vége
  ha DB > 0, akkor
    ki: ÖSSZEG / DB
  különben
    ki: "Az adatok között nem szerepelt egy fiú magassága sem"
  elágazás vége
eljárás vége
```


Kiválasztás tétele (általános)

- Adott az N elemű A vektor és egy X elem, melyről biztosan tudjuk, hogy eleme A -nak.
- Kérdés: hányas indexű eleme A -nak X ?

```
eljárás kiválasztás  
  I = 0  
  ciklus, amíg A[I] ≠ X  
    I = I + 1  
  ciklus vége  
  ki: I  
eljárás vége
```

Mi legyen a ciklusfeltétel, ha nem tudjuk, hogy benne van-e az X az A vektóban? Ciklus után hogy tudjuk meg az eredményt?

Szélsőérték-kiválasztás tétele

- Minimumkiválasztás, maximumkiválasztás

```
eljárás maximumkiválasztás1  
    MAXINDEX = 0  
    ciklus I = 1-től (N - 1)-ig  
        ha A[I] > A[MAXINDEX], akkor  
            MAXINDEX = I  
    elágazás vége  
    ki: MAXINDEX  
eljárás vége
```

...

7	9	2	3	9	5	9	3	4	7
---	---	---	---	---	---	---	---	---	---

(Hand-drawn red arrows point from the first, fourth, and sixth '9' to the MAXINDEX position in the algorithm above.)

Szélsőérték kiválasztás példa

- A megszámlálás tételénél megadott adatokat figyelembe vételével keressük a legmagasabb lány magasságát!

- Írjuk meg Pythonban!

```
eljárás példa4
```

```
MAX = -∞
```

```
MAXINDEX = -1
```

```
ciklus I = 0-tól 14-ig
```

```
    ha NEM[I]='N' és MAGASSÁG[I] > MAX, akkor
```

```
        MAX = MAGASSÁG[I]
```

```
        MAXINDEX = I
```

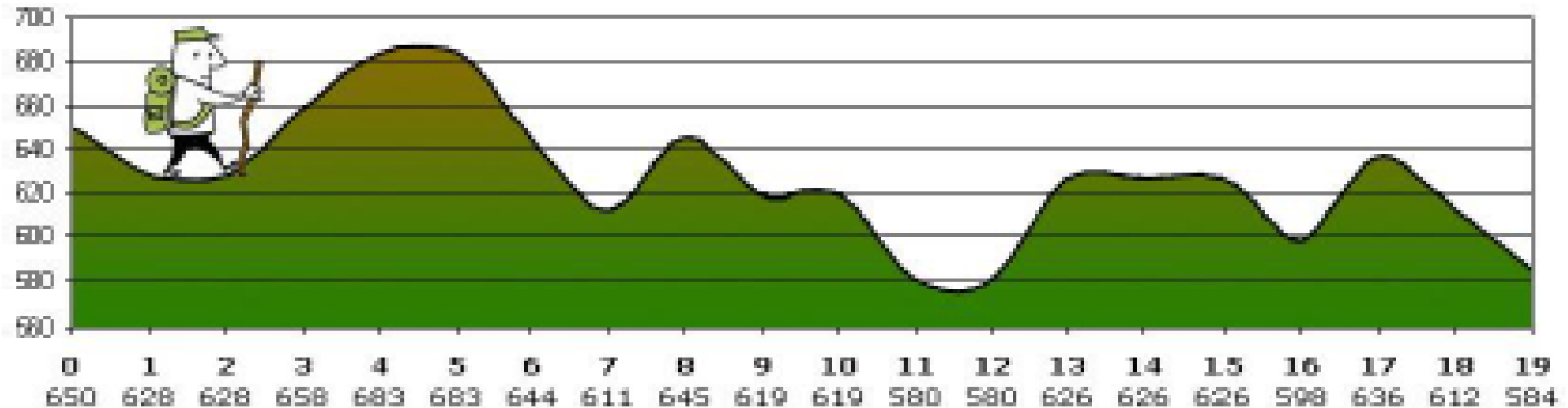
```
ciklus vége
```

```
ki: MAGASSÁG[MAXINDEX]
```

```
eljárás vége
```

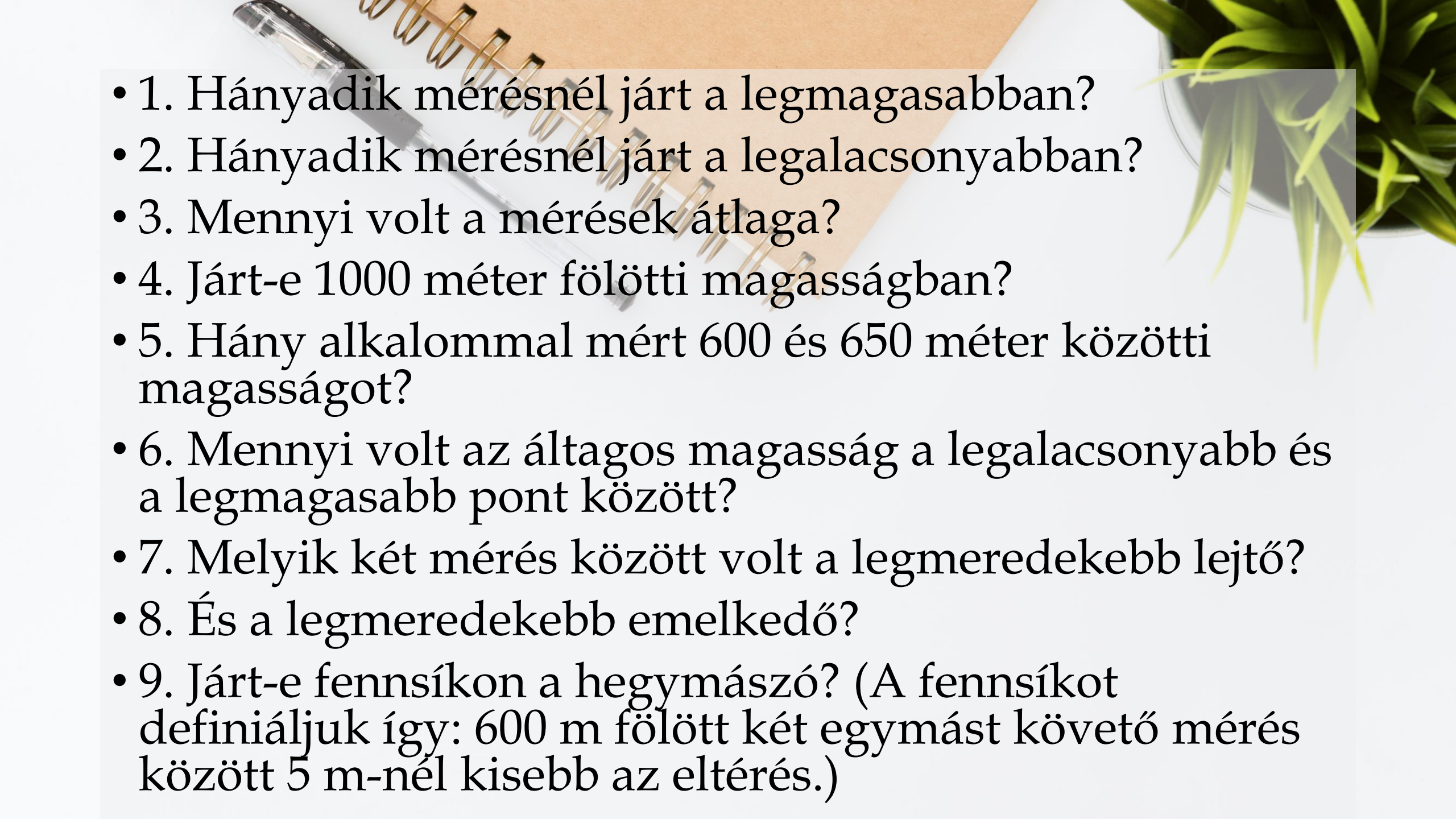

Önálló feladat Hegymászó

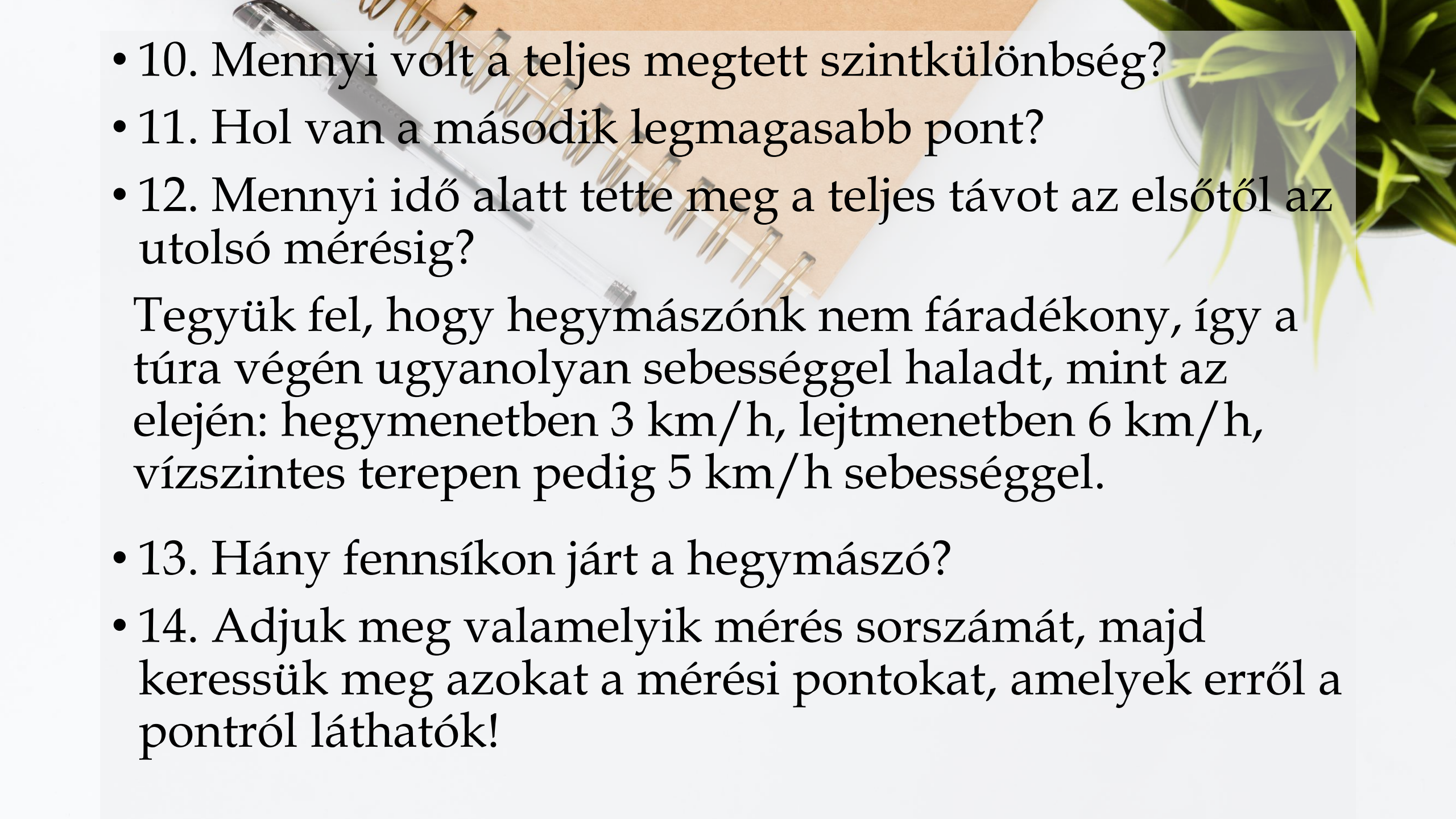
Egy hegymászó 20 kilométeres túrát tett, és kilométerenként megmérte az tengerszint feletti magasságot. Az alábbi értékeket rögzítette az ADATOK vektorban:



ADATOK[]

650	628	658	683	683	683	644	611	645	619	619	580	580	626	626	626	598	636	612	584
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

- 
- 1. Hányadik mérésnél járt a legmagasabban?
 - 2. Hányadik mérésnél járt a legalacsonyabban?
 - 3. Mennyi volt a mérések átlaga?
 - 4. Járt-e 1000 méter fölötti magasságban?
 - 5. Hány alkalommal mért 600 és 650 méter közötti magasságot?
 - 6. Mennyi volt az átlagos magasság a legalacsonyabb és a legmagasabb pont között?
 - 7. Melyik két mérés között volt a legmeredekebb lejtő?
 - 8. És a legmeredekebb emelkedő?
 - 9. Járt-e fennsíkon a hegymászó? (A fennsíkot definiáljuk így: 600 m fölött két egymást követő mérés között 5 m-nél kisebb az eltérés.)

- 
- 10. Mennyi volt a teljes megtett szintkülönbség?
 - 11. Hol van a második legmagasabb pont?
 - 12. Mennyi idő alatt tette meg a teljes távot az elsőtől az utolsó mérésig?

Tegyük fel, hogy hegymászónk nem fáradékony, így a túra végén ugyanolyan sebességgel haladt, mint az elején: hegymenetben 3 km/h, lejtmenetben 6 km/h, vízszintes terepen pedig 5 km/h sebességgel.

- 13. Hány fennsíkon járt a hegymászó?
- 14. Adjuk meg valamelyik mérés sorszámát, majd keressük meg azokat a mérési pontokat, amelyek erről a pontról láthatók!



Köszönöm a figyelmet!

Felhasznált szakirodalom

- Tömösközi Péter: Algoritmizálás alapjai EKF, 2011. ISBN 978-615-5221-14-9 (<https://docplayer.hu/10208397-Algoritmizalas-alapjai-tomoskozi-peter.html>)
- Dr. Geda Gábor: Adatszerkezetek és algoritmusok, EKF TTK, 2013 (<https://docplayer.hu/17242165-Adatszerkezetek-es-algoritmusok-geda-gabor.html>)
- Dr. Juhász István: Adatszerkezetek és algoritmusok jegyzet (<https://docplayer.hu/3509405-Adatszerkezetek-es-algoritmusok.html>)