21. Lineáris keresés

A lineáris keresés algoritmusa

A lineáris keresés algoritmusa megkeresi a sorozat (tömb) egy meghatározott tulajdonságú elemét. Keresünk például egy megadott nevű diákot, vagy egy olyan diákot, aki magasabb egy beolvasott értéknél stb. A keresés eredményeként az elem indexét, esetleg az értékét adjuk meg. Jelezzük azt is, ha nem találtunk megfelelő tulajdonságú elemet.

Az algoritmusban a tömb elejétől kezdve egy feltételes ciklussal megvizsgáljuk az elemeket mindaddig, amíg rá nem találunk a keresett értékre. Az aktuális indexet *I*-vel jelöljük. Kezdőértéke megegyezik a tömb kezdőindexével. A ciklusfejben megnézzük, hogy az *I*-edik elem rendelkezik-e a keresett tulajdonsággal. Ha nem, akkor továbblépünk a következő elemre:

```
I = Kezdőindex
CIKLUS AMÍG Tömb(I) nem adott tulajdonságú
   I = I+1
CIKLUS VÉGE
```

Ha a tömb nem tartalmaz egyetlen keresett tulajdonságú elemet sem, akkor a fenti ciklus az indexhatár túllépéséhez vezet, ami futási hibát okoz. Ezért az ismétlési feltételt kiegészítjük az indexhatár vizsgálatával:

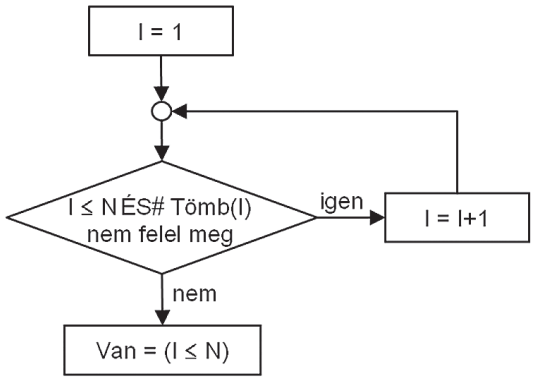
```
CIKLUS AMÍG I ≤ Végsőindex ÉS Tömb(I) nem adott tulajdonságú
   I = I+1
CIKLUS VÉGE
```

Ha a *Végsőindex* a tömb utolsó elemére mutat, akkor az indexhatár túllépését még így sem akadályoztuk meg. Mi történik ugyanis, ha nem találtunk megfelelő elemet, és az *I* elérte a végső index értékét? Belépünk a ciklusmagba, megnöveljük az *I*-t, majd ismét elvégezzük az ismétlési feltétel vizsgálatát. Így az ÉS-művelet második operandusában már nem létező (*Végsőindex*+1 indexű) tömbelemre hivatkozunk. Ennek elkerüléséhez logikai rövidzárat alkalmazunk:

```
CIKLUS AMÍG I ≤ Végsőindex ÉS# Tömb(I) nem adott tulajdonságú
   I = I+1
CIKLUS VÉGE
```

Mivel az ÉS-művelet első operandusa hamis, a második operandus kiértékelésére nem kerül sor. Figyeljünk a ciklusfejben szereplő feltételek sorrendjére! Mi történne, ha először a *Tömb*(I)-t vizsgálnánk meg, és csak utána az *I* értékét?

Ha a programozási nyelv nem teszi lehetővé a logikai műveletek rövidre zárását, akkor például egy további, valós adatot nem tartalmazó tömbelem felvételével kerülhetjük el az indexhatár túllépését.



A lineáris keresés folyamatábrája

A ciklus kétféleképpen fejeződhet be. Ha találtunk megfelelő elemet, akkor az *I* értéke legfeljebb a végső indexszel egyezik meg. Így a ciklus után ellenőrizni tudjuk, hogy sikeres volt-e a keresés:

```
HA I ≤ Végsőindex AKKOR
   Ki: I ' a megtalált elem indexe
EGYÉBKÉNT
   Ki: nem találtunk megfelelő elemet
ELÁGAZÁS VÉGE
```

Egy nagyobb program részeként az is elképzelhető, hogy nem kiírással végződik a ciklus. Ekkor a keresés sikerét például egy logikai változóval jelezhetjük:

```
Van = (I ≤ Végsőindex)
HA Van AKKOR
   Melyik = I
ELÁGAZÁS VÉGE
```

```
I = Kezdőindex
CIKLUS AMÍG I ≤ Végsőindex ÉS# Tömb(I) nem adott tulajdonságú
   I = I+1
CIKLUS VÉGE
Van = (I ≤ Végsőindex)
HA Van AKKOR
   Melyik = I
ELÁGAZÁS VÉGE
```

A lineáris keresés algoritmusa

1. gyakorlat. Keressünk meg az *Adat* kódfájlban egy beolvasott nevet, majd írjuk ki a diák magasságát! Jelezzük azt is, ha nem szerepel a megadott név a tömbben!

Az *Array.IndexOf(Tömbnév, kifejezés,* [, *kezdőindex* [, *darab*]]) osztálymetódus megadja az első olyan elem indexét, melynek értéke megegyezik a kifejezés értékével. Ha nincs ilyen elem, akkor a visszatérési érték –1. A *LastIndexOf* osztálymetódus a tömb végéről kezdi a keresést (a legutolsó elemet találja meg).

A lineáris keresés tulajdonságai

A lineáris keresés sorra egymás után megvizsgálja a tömb elemeit, így a végrehajtási idő a tömbelemek számával arányos. Erre utal az algoritmus elnevezése. Egyenletes eloszlású adatoknál átlagosan a tömb felét kell végignézni. N elem esetén tehát átlagosan $N/2$ összehasonlítást végzünk.

Ha több elem is rendelkezik az adott tulajdonsággal, akkor a lineáris keresés a legkisebb indexűt határozza meg közülük. Természetesen a keresést a tömb végétől visszafelé is végezhetjük. Így a legnagyobb indexű elemre akadunk rá először.

2. gyakorlat. Módosítsuk az 1. gyakorlat programját úgy, hogy visszafelé keressen a tömbben!

Lineáris keresés rendezett sorozatban

Ha adataink nagyság szerint vagy ábécérendben helyezkednek el, akkor csökkenthetjük az összehasonlítások számát. Így gyorsabb és hatékonyabb algoritmust kapunk. Amint a keresett értéknél nagyobb elemhez érkeztünk, fölösleges tovább folytatnunk a keresést, nem találtunk megfelelő elemet:

```
CIKLUS AMÍG I ≤ Végsőindex ÉS# Tömb(I) < keresett érték
        I = I+1
CIKLUS VÉGE
Van = (I ≤ Végsőindex ÉS# Tömb(I) = keresett érték)
```

Miért egészítettük ki a találatot jelző logikai kifejezést a *Tömb*(I) vizsgálatával?

3. gyakorlat. Írjunk programot, amely megvizsgálja, hogy szerepel-e egy beolvasott név a *Nevek* kódfájlban! Alkalmazzuk a rendezett sorozatban történő keresés algoritmusát!

Feladatok

1. Módosítsuk a lineáris keresés ciklusát úgy, hogy ismétlési feltétel helyett kilépési feltételt alkalmazunk! Hogyan írhatjuk át az ismétlési feltételt kilépési feltételre a de Morgan-azonosságok segítségével?