

Programozási tételek

- **Egy sorozathoz egy érték hozzárendelése**
 - Összegzés tétele

Adott egy N elemű számsorozat: $A(N)$. Számoljuk ki az elemek összegét!

```
Eljárás:
    S:=0
    Ciklus I=1-től N-ig
        S:=S+A(I)
    Ciklus vége
Eljárás vége.
```

- Eldöntés tétele

N elemű sorozat és egy a sorozaton értelmezett T tulajdonság. Van-e a sorozatnak legalább egy T tulajdonságú eleme?

```
Eljárás:
    I:=1
    Ciklus amíg I<=N és A(I) nem T tulajdonságú
        I:=I+1
    Ciklus vége
    VAN:=I<=N
Eljárás vége
```

("VAN" egy logikai változó, amely akkor és csak akkor igaz, ha $I \leq N$)

Hasonló feladat: igaz-e, hogy a sorozat minden eleme T tulajdonságú?

```
Eljárás:
    I:=1
    Ciklus amíg I<=N és A(I) T tulajdonságú
        I:=I+1
    Ciklus vége
    IGAZ:=I>N
Eljárás vége
```

- Kiválasztás tétele

Adott egy N elemű sorozat, egy - a sorozat elemein értelmezett - T tulajdonság, és tudjuk, hogy a sorozatban van legalább egy T tulajdonságú elem. A feladat ezen elem sorszámának meghatározása.

```

Eljárás:
    I:=1
    Ciklus amíg A(I) nem T tulajdonságú
        I:=I+1
    Ciklus vége
    SORSZ:=I
Eljárás vége

```

- Megszámlálás tétele

Adott egy N elemű sorozat és egy - a sorozat elemein értelmezett - T tulajdonság. Feladat a T tulajdonsággal rendelkező elemek megszámlálása.

```

Eljárás:
    S:=0
    Ciklus I=1-től N-ig
        Ha A(I) T tulajdonságú akkor S:=S+1
    Ciklus vége
Eljárás vége.

```

- keresések
 - Lineáris keresés tétele

Általános feladat: N elemű sorozat; sorozat elemein értelmezett T tulajdonság. Van-e T tulajdonságú elem és ha van, akkor mi a sorszáma. (Eldöntés és kiválasztás együtt.)

```

Eljárás:
    I:=1
    Ciklus amíg I<=N és A(I) nem T tulajdonságú
        I:=I+1
    Ciklus vége
    VAN:=I<=N
    Ha VAN akkor SORSZ:=I
Eljárás vége.

```

- Logaritmikus keresés tétele

Általános feladat: N elemű rendezett sorozat; egy keresett elem (X). Szerepel-e a keresett elem a sorozatban és ha igen, akkor mi a sorszáma.

Kihasználjuk, hogy a sorozat rendezett, így el tudjuk dönteni, hogy a keresett elem az éppen vizsgált elemhez képest hol helyezkedik el.

Al, F: intervallum alsó és felső végpontjai.

```

Eljárás:
  Al:=1
  F:=N
  Ciklus
    K:=INT((Al+F)/2)
    Ha A(K)<X akkor Al:=K+1
    Ha A(K)>X akkor F:=K-1
  amíg Al<=F és A(K)≠ X      (amíg Al>F vagy A(K)=X)
  Ciklus vége
  VAN:=Al<=F
  Ha VAN akkor SORSZ:=K
Eljárás vége.

```

Megjegyzések:
 azért hívják logaritmikus keresésnek, mert a ciklus lépésszáma kb.
 $\log N$
 sokkal hatékonyabb rendezett sorozatra, mint a lineáris keresés

- o Maximum kiválasztás tétele

Sorozat legnagyobb elemének indexe.

```

Eljárás:
  INDEX:=1
  Ciklus I=2-től N-ig
    Ha A(INDEX) < A(I) akkor INDEX:=I
  Ciklus vége
Eljárás vége.

```

- o Minimumkiválasztás tétele

Sorozat legkisebb elemének indexe.

```

Eljárás:
  ÉRTÉK:=A(1)
  Ciklus I=2-től N-ig
    Ha A(I) < ÉRTÉK akkor ÉRTÉK:=A(I)
  Ciklus vége
Eljárás vége.

```

- **Egy sorozathoz egy sorozat hozzárendelése**

- Kiválogatás tétele

Egy N elemű sorozat összes T tulajdonságú elemét kell meghatározni. A kiválogatott elemek sorszámainak egy $B()$ vektorban gyűjtjük.

Eljárás:

$J := 0$

Ciklus $I = 1$ -től N -ig

Ha $A(I)$ T tulajdonságú, akkor $J := J + 1$

$B(J) := I$

Ciklus vége

Eljárás vége.

- Rendezések (rendezési szempontok)

Rendezési eljárás kiválasztásánál szempontok:

- tárigény
- végrehajtási idő
- összehasonlítások, mozgások, cserék száma
- adott gépi környezet.

Hatékonysági mutatók:

Tárigény: rendezendő adatok száma legyen.

Összehasonlítások száma: azon relációk, ahol egyik oldalon rendezendő adat áll.

Mozgások száma: azon értékadó utasítások, melyekben legalább az egyik oldalon rendezendő adat áll.

Végrehajtási idő: mért idő egy konkrét programmal.

- Rendezés közvetlen kiválasztással

Rendezés közvetlen kiválasztással

Módszer lényege:

Rendezendő számok az A vektor elemei.

Első menetben az $A(1)$ -et összehasonlítjuk az összes elemmel és ha kisebbet találunk nála, akkor felcseréljük.

Így az első menet végére a legkisebb elem lesz az első helyen.

Ezután ezt ismétljük az $A(2)$ -es elemmel, stb.

$N-1$ menet után rendezett lesz a sorozat.

```

Eljárás:
    Ciklus I=1-től N-1-ig
        Ciklus J=I+1-től N-ig
            Ha A(J) < A(I) akkor C:=A(J)
            A(J):=A(I)
            A(I):=C
        Ciklus vége
    Ciklus vége
Eljárás vége.

```

Hatékonysági mutatók:

Tárigény: $N+1$

Összehasonlítások száma: $N*(N-1)/2$

Mozgatások száma: 0-tól $3*N*(N-1)/2$ -ig lehetséges

Végrehajtási idő: 2980 s ($N=500$)

Rendezés minimum-kiválasztással

Módszer lényege: Felesleges cserék kiküszöbölése érdekében két segédváltozót vezetünk be (legkisebb elem értékének és indexének). .

```

Eljárás:
    Ciklus I=1-től N-1-ig
        INDEX:=I
        ÉRTÉK:=A(I)
        Ciklus J=I+1-től N-ig
            Ha A(J)<ÉRTÉK akkor ÉRTÉK:=A(J)
            INDEX:=J
        Ciklus vége
        A(INDEX):=A(I)
        A(I):=ÉRTÉK
    Ciklus vége
Eljárás vége.

```

Hatékonysági mutatók:

Tárigény: $N+1$

Összehasonlítások száma: $N*(N-1)/2$

Mozgatások száma: $3*(N-1)$ -től $3*(N-1)+(N*N/4)$ -ig lehetséges

Végrehajtási idő: 1650 s ($N=500$)

Buborékos rendezés:

Módszer lényege:

Hasonlítsuk össze a vektor első és második elemét, s ha az első nagyobb, cseréljük meg őket. Ezt követően hasonlítsuk össze a vektor második és harmadik elemét, s ha a második

nagyobb, cseréljük meg őket, és így tovább. Látható, hogy ekkor a vektor legnagyobb eleme biztosan a helyére kerül, még akkor is, ha ő volt az első. Sajnos ez a többi elemre nem feltétlenül teljesül, tehát ahhoz, hogy a második legnagyobb elem a helyére kerüljön, ismét el kell indulnunk a vektor elejéről. Az első elemet össze kell hasonlítani a másodikkal, a másodikat a harmadikkal, stb., azonban most már elég elmenni az utolsó előtti elemig, hiszen az utolsó a helyén van. A következő körben már a harmadik legnagyobb elemet tesszük a helyére, és így tovább. A külső ciklust tehát annyiszor kell lefuttatnunk, ahány eleme van a vektorunknak.

```
Eljárás:
    Ciklus I=1-től N-ig
        Ciklus J=1-től N-I-ig
            Ha A(J)>A(J+1) akkor Cs:=A(J)
            A(J):=A(J+1)
            A(J+1):=Cs
        Ciklus vége
    Ciklus vége
Eljárás vége.
```

Hatékonysági mutatók:

Tárigény: $N+1$

Összehasonlítások száma: $N*(N-1)/2$

Mozgatások száma: 0-tól $3*N*(N-1)/2$ -ig lehetséges

Végrehajtási idő: 3620 s ($N=500$)

Egyszerű beillesztéses rendezés

Módszer lényege: Mintha kártyáinkat egyesével felvéve sorba raknánk. ($N-1$ menet)

```
Eljárás:
    Ciklus J=2-től N-ig
        I:=J-1
        A:=A(J)
        Ciklus amíg I > 0 és A < A(I)
            A(I+1):=A(I)
            I:=I-1
        Ciklus vége
        A(I+1):=A
    Ciklus vége
Eljárás vége.
```

Hatékonysági mutatók:

Tárigény: $N+1$

Összehasonlítások száma: $N-1$ -től $N*(N+1)/2-1$ -ig változhat

Mozgatások száma: $2 \cdot N - 1$ -től $2 \cdot (N - 1) + N \cdot (N - 1) / 2$ -ig lehetséges
Végrehajtási idő: 1950 s ($N=500$)

Metszetképzés

Általános feladat: Rendezésünkre áll egy N és egy M elemű halmaz az $A()$ és a $B()$ vektorban ábrázolva. Készítsük el a két halmaz metszetét a $C()$ vektorba!

```
Eljárás:
    CN:=0
    Ciklus I=1-től N-ig
        J:=1
        Ciklus amíg J<=M és A(I)<>B(J)
            J:=J+1
        Ciklus vége
        Ha J<=M akkor CN:=CN+1
        C(CN):=A(I)
    Ciklus vége
Eljárás vége.
```

Únióképzés

Általános feladat: Rendezésünkre áll egy N és egy M elemű halmaz az $A()$ és a $B()$ vektorban ábrázolva. Készítsük el a két halmaz egyesítését a $C()$ vektorba!

```
Eljárás:
    Ciklus I=1-től N-ig
        C(I):=A(I)
    Ciklus vége
    CN:=N
    Ciklus J=1-től M-ig
        I:=1
        Ciklus amíg I<=N és A(I)<>B(J)
            I:=I+1
        Ciklus vége
        Ha I>N akkor CN:=CN+1
        C(CN):=B(J)
    Ciklus vége
Eljárás vége.
```

Összefuttatás

Általános feladat: Két rendezett sorozat uniója úgy, hogy a rendezettség megmaradjon.

```
Eljárás:
    I:=1
    J:=1
    K:=0
    Ciklus amíg I<=N és J<=M
        K:=K+1
        Elágazás
            A(I) < B(J) esetén
                C(K):=A(I)
                I:=I+1
            A(I) = B(J) esetén
                C(K):=A(I)
                I:=I+1
                J:=J+1
            A(I) > B(J) esetén
                C(K):=B(J)
                J:=J+1
        Elágazás vége
    Ciklus vége
    Ciklus amíg I<=N
        K:=K+1
        C(K):=A(I)
        I:=I+1
    Ciklus vége
    Ciklus amíg J<=M
        K:=K+1
        C(K):=B(J)
        J:=J+1
    Ciklus vége
Eljárás vége.
```