

# PROGRAMOZÁSMÓDSZERTAN

## 5. ELŐADÁS'2004

(VÁZLAT)

### 1. RENDEZÉSI TÉTELEK

#### 1.1. Rendezési tételekről előzetesen

##### 1.1.1. A feladat

Egy adatsorozat elemeinek átrendezése úgy, hogy elemei a művelet után valamely „szempontból” rendezettek legyenek.

A **szempontot** gyakorta az elem egy valamely része, **mezője** testesíti meg: a **kulcs**.

Azaz az Elem=**Kulcs**×Érték. Ekkor  $e_1 \leq e_2 \stackrel{def}{\Leftrightarrow} e_1.\text{Kulcs} \leq e_2.\text{Kulcs}$ .

E helyett elképzelhető egy, az elemek alaphalmazán értelmezett olyan **függvény** is, amely értékészlete egy nyilvánvalóan rendezett halmaz, pl.  $\mathbb{N}$ .

Azaz **KulcsFv**:Elem $\rightarrow\mathbb{N}^1$ . Ekkor  $e_1 \leq e_2 \stackrel{def}{\Leftrightarrow} \text{KulcsFv}(e_1) \leq \text{KulcsFv}(e_2)$

##### 1.1.2. Fajtái

- Belső rendezések – a rendezendő sorozat **befér a memóriába**, ezért feltehetjük, hogy **tömbben** tárolható.
- Külső rendezések – a sorozat **egésze nem** tartható egyidejűleg a tárban, ekkor elemei valamely (háttértár) **állomány(ok)**ban tárolódnak. (L. Adatfeldolgozás előadásban.)

*Megjegyzés:*

Ha –bár nem fér a sorozat a memóriába, de– olyan háttértár adatszerkezetben tárolódik, amely **közvetlen elérésű**, vagyis létezik egy rajta értelmezett **indexszelés**-szerű művelet (az elemeléréshez, illetve az elemmódosításhoz), akkor kicsi átalakítással a belső rendezés algoritmusai alkalmazhatók.

Ebből a megjegyzésből is érződik a fenti fajták elnevezésének nem túl szerencsés volta. Kifejezőbb lenne valami efféle: „Rendezés **közvetlen elérésű** adatszerkezetben”, ill. „Rendezés **szekvenciális** adatszerkezetben”.

<sup>1</sup> Persze  $\mathbb{N}$  helyett megfelelő bármely „közismert” rendezett halmaz. Pl.:  $\mathbb{R}$ ,  $\mathbb{Z}$ ,  $\mathbb{S}$  (=Szövegek halmaza) stb.

### 1.1.3. Algoritmikus „filozófiák”

Mivel a rendezés során *permutáljuk* a bemenő sorozatot, sok út vezet a rendezettből rendezett felé, ezért sokféle módszer található a permutálásra. Ezek eltérő **hatékonysággal** rendelkeznek, más körülmények között másképpen („okosan”, „bután”) viselkednek.

Fajta	Lényeg
<i>Transzpozíció</i>	Ha az összehasonlításban szereplő <b>elem</b> <i>párok</i> egymáshoz képesti sorrendje (transzpozíció) rossz, akkor megcseréljük... Pl.: (4,3,2,5,1) $\Rightarrow$ (3,4,2,5,1) $\Rightarrow$ (3,2,4,5,1) $\Rightarrow$ (3,2,4,5,1) $\Rightarrow$ (3,2,4,1,5) $\Rightarrow$ (2,3,4,1,5) $\Rightarrow$ (2,3,4,1,5) $\Rightarrow$ (2,3,1,4,5) $\Rightarrow$ ...
<i>Szelekció</i>	Mindig a <b>rendezettség szerinti következő</b> kiválasztása (szelekciója), majd helyre tétele csere útján... Pl.: (4,3,2,5,1) $\Rightarrow$ ... minimum kiválasztás(4,3,2,5,1): 1 ... $\Rightarrow$ (1,3,2,5,4) $\Rightarrow$ ... minimum kiválasztás(3,2,5,4): 2 ... $\Rightarrow$ (1,2,3,5,4) $\Rightarrow$ ...
<i>Beszűrés</i>	A következő elemnek a rendezettségnek megfelelő helyre tétele a már <b>rendezett részsorozat</b> ban... Pl.: (4,3,2,5,1) $\Rightarrow$ (3,4,2,5,1) $\Rightarrow$ (2,3,4,5,1) $\Rightarrow$ (2,3,4,5,1) $\Rightarrow$ (1,2,3,4,5)

### 1.1.4. A hatékonyságról dióhéjban

#### Mit mérhetünk?

- Helyfoglalás
- Végrehajtási idő

#### A mérés „skálája”:

- Minimális – optimista
- Átlagos – praktikus
- Maximális – pesszimista

Az absztrakt mérés matematikai eszközeiről:<sup>2</sup>

Miként változik az **elemszám**mal (sorozathosszal) a hasonlítószám, ill. a mozgatószám; társzükséglet *aszimptotikusan*?

Egy  $\mathbb{N} \rightarrow \mathbb{N}$  függvény aszimptotikus<sup>3</sup> viselkedésének kifejezésére (többek közt) a  $\Theta$ - (theta),  $O$ - (nagy ordó),  $\Omega$ -függvények alkalmasak:

$$\Theta(g(n)) := \{f(n) \mid \exists c_1, c_2 \in \mathbb{R}_+, n_0 \in \mathbb{N}: \forall n \geq n_0 \Rightarrow c_1 * g(n) \leq f(n) \leq c_2 * g(n)\} - \text{korlátos}$$

$$O(g(n)) := \{f(n) \mid \exists c \in \mathbb{R}_+, n_0 \in \mathbb{N}: \forall n \geq n_0 \Rightarrow 0 \leq f(n) \leq c * g(n)\} - \text{felülről korlátos}$$

$$\Omega(g(n)) := \{f(n) \mid \exists c \in \mathbb{R}_+, n_0 \in \mathbb{N}: \forall n \geq n_0 \Rightarrow c * g(n) \leq f(n)\} - \text{alulról korlátos}$$

<sup>2</sup> Ajánlható ehhez az alábbi irodalom:

- Cormen et al.: Algoritmuskönyv, Műszaki Könyvkiadó, 1997 [18-34. oldalak]
- Knuth: A számítógép-programozás művészete 3., Műszaki Könyvkiadó, 1988

<sup>3</sup> Azért éppen  $\mathbb{N} \rightarrow \mathbb{N}$  függvény, mert a feladat szempontjából érdekes sorozat *hossza* függvényében szeretnénk meghatározni bizonyos gyakoriságszerű programjellemzők tendenciáját (ezért aszimptotikusan).

Megjegyzések:

1.  $f(n)=O(g(n))$  helytelen szokás, hisz  $O(\dots)$  függvényhalmaz. Helyesen:  $f(n)\in O(g(n))$
2. Értelme persze csak akkor van a fentieknek, ha az  $f(n)$  függvény valamilyen „egyszerű” függvény; pl. *hatványfüggvény*, *exponenciális függvény* stb.

Pl.: ha  $f(n)=A*n^2+B*n+C$ , akkor

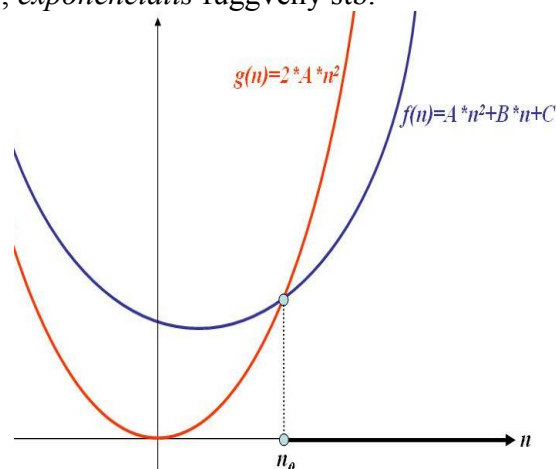
$$f\in O(n^2),$$

hiszen  $c:=2*A$ , s ekkor megadható

egy megfelelő választás  $n_0$ -ra

$$n_0 := \left\lceil \frac{B}{2*A} + \sqrt{\frac{B^2}{4*A^2} + C} \right\rceil$$

hogy  $n\geq n_0$  esetén  $0\leq f(n)\leq c*n^2$  teljesüljön. Sőt az  $f\in\Theta(n^2)$  is igaz.



### 1.1.5. A specifikációjukról

Legtöbbjük specifikációja ugyanaz:

**BelsőRendezés( $H^*$ ,  $\mathcal{F}(H\times H, \mathbb{L})$ ):  $H^*$**

Megjegyzések:

Be:  $N\in\mathbb{N}$ ,  $X\in H^*$ ,  $H=\text{Kulcs}\times\text{Érték}$ ,  
 $\leq_{\text{Kulcs}}: \text{Kulcs}\times\text{Kulcs}\rightarrow\mathbb{L}$

Sorozat és egy reláció.

Ki:  $X'\in H^*$

A transzformáció helyben történik.

Ef:  $N=\text{Hossz}(X) \wedge \leq_{\text{Kulcs}}$  rendezés  $\wedge$   
 $X$  közvetlen elérésű

Rendezési reláció a szokásos tulajdonságokkal;  $X$  elemei indexeléssel elérhetők:  $\exists \text{Elem}(x,i)=x_i$  operáció....

Uf:  $X'\in \underline{P}(X) \wedge \text{RendezettSorozat}_{\leq}(X')$

A kimenet a bemenet egy permutációja, amelyben az elemek rendezetten követik egymást. A továbbiakban is fölteszük, hogy a rendezettség mindig **nem-csökkenő**t jelent.

A tételek szerkezete visszatérő:

- *specifikáció* (amit elhagyunk, ha a fentivel megegyező),
- *algoritmus*, amelyben az alábbi definíciós rész unós untilg ismétlődik (ezért ezt sem írjuk le):

**Konstans** MaxN: Egész (???)

**Típus** THk=Tömb(1..MaxN:TH) [TH az alaphalmaz típusa]

**Infex Operátor**  $\leq$ (Konstans e, f:TH): Logikai

**Másként** KisebbEgyenlő

KisebbEgyenlő:=e.Kulcs≤f.Kulcs [visszavezetés  $\leq_{\text{Kulcs}}$ -ra]

**Operátor vége.**

**Infex Operátor**  $>$ (Konstans e, f:TH): Logikai

**Másként** Nagyobb

Nagyobb:=nem(e≤f)

**Operátor vége.**

- *hatékonysági megfontolások.*

## 1.2. Belső rendezések

### 1.2.1. Egyszerű cserés rendezés

**CsereRendezés**( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$

Alg:

```

Eljárás CsereRendezés (Konstans N:Egész, Változó4 X:THk) :
  Változó i, j:Egész
  Ciklus i=1-től N-1-ig
    Ciklus j=i+1-től N-ig
      Ha X(i)>X(j) akkor Csere(X(i),X(j))
    Ciklus vége
  Ciklus vége
Eljárás vége.

Eljárás Csere(Változó e, f:TH) :5
  Változó s:TH
  s:=e; e:=f; f:=s
Eljárás vége.

```

A filozófiát tekintve: *transzpozíciós* rendezés.

*Hatékonyság:*

<i>Szempont</i>	<i>Min</i>	<i>Max</i>	
<b>Helyfoglalás</b>	$N+1$ <sup>6</sup>		$\Theta(N)$
<b>Hasonlításszám</b>	$N*(N-1)/2$		$\Theta(N^2)$
<b>Mozgatásszám</b>	0	$3*N*(N-1)/2$	$O(N^2)$

Érdeemes megfontolni: milyen tulajdonságú bemenő adat esetén viselkedik a fentiek szerint.

### 1.2.2. Minimumkiválasztásos rendezés

**Minimumkiálasztásos**( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$

Alg:

```

Eljárás MinimumKiválasztásos (Konstans N:Egész
  Változó X:THk) :
  Változó i, j:Egész

```

<sup>4</sup> Ki- és bemenő paraméter egyben, ezért változó.

<sup>5</sup> A csere-eljárás később is elő fog bukkanni, de akkor már nem deklaráljuk újra.

<sup>6</sup> "A+1" az s TH-típusú segédváltozó.

```

Ciklus i=1-től N-1-ig
  [mini:=MinimumKiválasztás(X(i..N),>)]
  mini:=i
  Ciklus j=i+1-től N-ig
    Ha X(mini)>X(j) akkor mini:=j
  Ciklus vége
  Csere(X(i),X(mini))
Ciklus vége
Eljárás vége.

```

A filozófiát tekintve: szelekciós rendezés.

Hatékonyság:

Szempon	Min	Max	
Helyfoglalás	N+1		$\Theta(N)$
Hasonlításszám	$N*(N-1)/2$		$\Theta(N^2)$
Mozgatásszám	$3*(N-1)$		$\Theta(N)$

Gondoljon bele: mi lehet az *invariáns állítás* az egyes ciklusokban az egyszerű cserés és minnél a rendezésnél?

### 1.2.3. Buborékos rendezés

**Buborékos**( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$

Alg:

```

Eljárás Buborékos (Konstans N:Egész, Változó X:THk):
  Változó i,j:Egész
  Ciklus i=N-től 2-ig -1-esével
    Ciklus j=1-től i-1-ig
      Ha X(j)>X(j+1) akkor Csere(X(j),X(j+1))
    Ciklus vége
  Ciklus vége
Eljárás vége.

```

A filozófiát tekintve: transzpozíciós rendezés.

Hatékonyság:

Szempon	Min	Max	
Helyfoglalás	N+1		$\Theta(N)$
Hasonlításszám	$N*(N-1)/2$		$\Theta(N^2)$
Mozgatásszám	0	$3*N*(N-1)/2$	$O(N^2)$

Észreveheti, hatékonysági szempontból megegyezik az [egyszerű cserés rendezéssel](#). Meglepő ez?

### 1.2.4. Javított buborékos rendezés

**JavítottBuborékos( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$**

Alg:

```

Eljárás JavítottBuborékos (Konstans N:Egész
                                Változó X:THk) :
  Változó i, j, csH:Egész [csH:a csere helye, indexe]
  i:=N
  Ciklus amíg i≥2
    csH:=0
    Ciklus j=1-től i-1-ig
      Ha X(j)>X(j+1) akkor Csere(X(j),X(j+1)); csH:=j
    Ciklus vége
    i:=csH
  Ciklus vége
Eljárás vége.

```

A filozófiát tekintve: *transzpozíciós* rendezés.

A csH kezdőértékére tud-e még más, többé-kevésbé „logikus” választást?

*Hatékonyosság:*

<i>Szempont</i>	<i>Min</i>	<i>Max</i>	
Helyfoglalás	N+1		$\Theta(N)$
Hasonlításszám	N-1	$N*(N-1)/2$	$\Omega(N), O(N^2)$
Mozgatásszám	0	$3*N*(N-1)/2$	$O(N^2)$

A hatékonyság „átlagos” esetben javul.

### 1.2.5. Beillesztéses rendezés

**Beillesztéses( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$**

Alg:

```

Eljárás Beillesztéses (Konstans N:Egész, Változó X:THk) :
  Változó i, j:Egész
  Ciklus i=2-től N-ig
    j:=i-1
    Ciklus amíg j≥1 és X(j)>X(j+1)
      Csere(X(j),X(j+1)); j:-1
    Ciklus vége
  Ciklus vége
Eljárás vége.

```

A filozófiát tekintve: *beszúrásos* rendezés.

Hatékonyág:

Szempon	Min	Max	
Helyfoglalás	N+1		$\Theta(N)$
Hasonlításszám	N-1	$N*(N-1)/2$	$\Omega(N), O(N^2)$
Mozgatásszám	0	$3*N*(N-1)/2$	$O(N^2)$

Gondoljon bele: mik az *invariáns állítások* ennél a rendezésnél?

### 1.2.6. Javított beillesztéses rendezés

**JavítottBeillesztéses( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ):  $H^*$**

Alg:

```

Eljárás JavítottBeillesztéses (Konstans N:Egész
                                Változó X:THk) :
    Változó i, j:Egész
              segéd:TH
    Ciklus i=2-től N-ig
      j:=i-1; segéd:=X(i)
      Ciklus amíg j≥1 és X(j)>segéd
        X(j+1):=X(j); j:-1
      Ciklus vége
      X(j+1):=segéd
    Ciklus vége
Eljárás vége.

```

A filozófiát tekintve: *beszúrásos* rendezés.

Hatékonyág:

Szempon	Min	Max	
Helyfoglalás	N+1		$\Theta(N)$
Hasonlításszám	N-1	$N*(N-1)/2$	$\Omega(N), O(N^2)$
Mozgatásszám	$2*(N-1)$	$2*(N-1)+N*(N-1)/2$	$\Omega(N), O(N^2)$

### 1.3. Speciális rendezések

#### 1.3.1. Szétosztó rendezés

Egy **nem helyben** dolgozó rendezés, amely igen kemény elvárásokat fogalmaz meg a bemenő sorozatra.

**SzétosztóRendezés( $(KULCS \times H)^*, \mathcal{F}(KULCS \times KULCS, \mathbb{L})$ ):  $(KULCS \times H)^*$**

Be:  $N \in \mathbb{N}$ ,  $X \in (Kulcs \times H)^*$ ,  $Kulcs = \mathbb{N}$ ,  $\leq_{Kulcs}: Kulcs \times Kulcs \rightarrow \mathbb{L}$  –  $\leq_{Kulcs}$  elhagyható, hisz létezése nyilvánvaló

Ki:  $Y \in (Kulcs \times H)^*$  – nem helyben rendezés

Ef:  $N = Hossz(X) \wedge \leq_{Kulcs}$  rendezés  $\wedge X.Kulcs \in \mathcal{P}(\{1..N\})$  –  $\leq_{Kulcs}$  elhagyható, hisz tulajdonsága nyilvánvaló

Uf:  $Y \in \mathcal{P}(X) \wedge$  RendezettSorozat $_{\leq}(Y)$

Alg:

```

Eljárás SzétosztóRendezés (Konstans N:Egész, X:THk
                          Változó Y:THk) :
  Változó
    i, j :Egész
  Ciklus i=1-től N-ig
    Y(X(i).Kulcs) :=X(i)
  Ciklus vége
Eljárás vége.

```

Hatékonyság:

Szempon	Min	Max	
Helyfoglalás		2*N	$\Theta(N)$
Hasonlításszám		0	$O(1)$
Mozgatásszám		N	$\Theta(N)$

### 1.3.2. Számlálva szétosztó rendezés

Az előbbi rendezéstől abban tér el, hogy gyengíti az előfeltételt, és a tényleges rendezés előtt némi –kényszerű– előkészületeket tesz.

**Számlálva SzétosztóRendezés**((KULCS×H)<sup>\*</sup>, F(KULCS×KULCS, L)): (KULCS×H)<sup>\*</sup>

Be:  $N \in \mathbb{N}$ ,  $X \in (Kulcs \times H)^*$ ,  $M \in \mathbb{N}$ ,  $Kulcs = \mathbb{N} [\leq_{Kulcs} : Kulcs \times Kulcs \rightarrow \mathbb{L}]$

Ki:  $Y \in (Kulcs \times H)^*$

Ef:  $N = Hossz(X) \wedge \forall i \in [1..N]: x_i.Kulcs \in \{1..M\} [\wedge \leq_{Kulcs} \text{rendezés}]$

Uf:  $Y \in \mathcal{P}(X) \wedge \text{RendezettSorozat}_{\leq}(Y)$

Alg:

```

Eljárás Számlálva SzétosztóRendezés (Konstans N:Egész, X:THk
                          Változó Y:THk) :
  Konstans
    MaxM:Egész(???)
  Változó
    i:Egész
    Db:Tömb(1..MaxM:Egész)
  Db(1..M) :=0
  Ciklus i=1-től N-ig
    Db(X(i).Kulcs) :=+1
  Ciklus vége
  Ciklus i=2-től M-ig
    Db(i) :=Db(i-1) [Db(i): i. kulcsúak utolsójára mutat]
  Ciklus vége
  Ciklus i=1-től N-ig
    Y(Db(X(i).Kulcs)) :=X(i); Db(X(i).Kulcs) :=-1
  Ciklus vége
Eljárás vége.

```



Érdemes eltöprengeni, mely tételek bukkantak föl az algoritmusban!

Hatékonyág:

<i>Szempont</i>	<i>Min</i>	<i>Max</i>	
<b>Helyfoglalás</b>	2*N+ε*M		Θ(N)
<b>Hasonlításszám</b>	0		Θ(1)
<b>Mozgatásszám</b>	N		Θ(N)
<b>Kulcsmező indexelés száma</b>	3*N *		Θ(N)

A számláló tömb elemének mérete kicsiny, azaz „epszilon” (lehet) a rendezendő sorozat eleméhez képest. Mivel számosan lehetnek, helyet mégis foglalnak.

### 1.3.3. Számláló rendezés

Az alábbi rendezés sem helyben rendez, és több rendezésből is átvesz ötleteket.

#### SzámlálóRendezés( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$

Be:  $N \in \mathbb{N}, X \in H^*, \leq: H \times H \rightarrow \mathbb{L}$

Ki:  $Y \in H^*$

Ef:  $N = \text{Hossz}(X) \wedge \leq_H$  rendezés

Uf:  $Y \in \mathcal{P}(X) \wedge \text{RendezettSorozat}_{\leq}(Y)$

Alg:

**Eljárás SzámlálóRendezés (Konstans**  $N: \text{Egész}, X: \text{THk}$   
**Változó**  $Y: \text{THk}$ ):

**Változó**

$i, j: \text{Egész}$

$Db: \text{Tömb}(1.. \text{Max}N: \text{Egész})$

$Db(1..N) := 1$

[Majdnem [egyszerű cserés rendezés](#):]

**Ciklus**  $i=1$ -től  $N-1$ -ig

**Ciklus**  $j=i+1$ -től  $N$ -ig

**Ha**  $X(i) > X(j)$  **akkor**  $Db(i) := +1$

**különben**  $Db(j) := +1$

**Ciklus vége**

**Ciklus vége**

[[Szétosztó rendezés](#):]

**Ciklus**  $i=1$ -től  $N$ -ig

$Y(Db(i)) := X(i)$

**Ciklus vége**

**Eljárás vége.**

Az egyszerű cserés rendezést csak az inverziók felderítésére használjuk, az elemmozgatást a végére hagyjuk. Hogy miért?

\* Az  $X(i) . \text{Kulcs}$ -csal indexelést az inkrementálásban és a dekrementálásban csak egyszer számolva.

Hatékonyág:

Szempont	Min	Max	
Helyfoglalás	$2*N+\epsilon*N$		$\Theta(N)$
Hasonlításszám	$N*(N-1)/2$		$\Theta(N^2)$
Mozgatásszám	N		$\Theta(N)$
Kulcsmező indexelés száma	N		$\Theta(N)$

#### 1.4. Shell rendezés

Egy valóban hatékony rendezéssel fejezzük be a rendező algoritmusok sorát.

**ShellRendezés( $H^*, \mathcal{F}(H \times H, \mathbb{L})$ ): $H^*$**

Kezdjük egy beszúrásos rendezéses példával! A példa egy-egy sora a belső ciklus teljes lefutása utáni állapotot rögzíti. Ahol mozgás volt ott színes a háttér.

503	87	512	61	908	170	897	275	653	426	154	509	612	677	765	703
87	503	512	61	908	170	897	275	653	426	154	509	612	677	765	703
87	503	512	61	908	170	897	275	653	426	154	509	612	677	765	703
61	87	503	512	908	170	897	275	653	426	154	509	612	677	765	703
61	87	170	503	512	908	897	275	653	426	154	509	612	677	765	703
61	87	170	503	512	897	908	275	653	426	154	509	612	677	765	703
61	87	170	275	503	512	897	908	653	426	154	509	612	677	765	703
61	87	170	275	503	512	653	897	908	426	154	509	612	677	765	703
61	87	170	275	426	503	512	653	897	908	154	509	612	677	765	703
...															
61	87	154	170	275	426	503	509	512	612	653	677	703	765	897	908

Megállapítható, hogy nagyon lassan haladnak, nagyon sok apró lépéssel araszolnak a magas értékek a helyük felé. Ebből támadhat az ötlet: ne „egyesével” tekintsük sorozatnak a rendezendőt, hanem nagyobb „lépésközösével”. Pontosabban több részsorozatot rendezünk egyidejűleg, s „maj’ meklátyuk”... Nézzük pl. az előbbi 7-esével bontva részsorozatokra. Most az egy részsorozatba tartozókat azonos színű számokkal jelöltük.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
503	87	512	61	908	170	897	275	653	426	154	509	612	677	765	703
275	87	426	61	509	170	677	503	653	512	154	908	612	897	765	703

A „nagyot ugró” elemek: 275, 503, 426, 512, 509, 908, 677, 897. Összevethetjük az eredeti (1. sor) és a majdani végleges (3. sor) helyzetüket a pillanatnyival (5. sor). A 2. és a 4. sorban a végleges helyétől vett távolság látható.

503	87	512	61	908	170	897	275	653	426	154	509	612	677	765	703
6	0	6	3	11	2	8	3	2	4	8	4	3	2	1	3
61	87	154	170	275	426	503	509	512	612	653	677	703	765	897	908
4	0	3	3	3	2	4	1	2	1	8	4	3	1	1	3
275	87	426	61	509	170	677	503	653	512	154	908	612	897	765	703

A távolságösszeg jócskán csökkent: 66→43. (Mindezt 15 összehasonlítással értük el. Összevetve az előbbi példával: ott 15 összehasonlítás után jóval 50 fölött találnánk a távolságösszeget.)

Ha most már 5-ös rendezést végzünk:

275	87	426	61	509	170	677	503	653	512	154	908	612	897	765	703
154	87	426	61	509	170	677	503	653	512	275	908	612	897	765	703

Folytassuk 3-asával:

154	87	426	61	509	170	677	503	653	512	275	908	612	897	765	703
61	87	170	154	275	426	512	503	653	612	509	765	677	897	908	703

Majd 1-esével:

61	87	170	154	275	426	512	503	653	612	509	765	677	897	908	703
61	87	154	170	275	426	503	509	512	612	653	677	703	765	897	908

Fontosnak látszó kérdés: *hogyan változzék a lépésköz.*

Szemponatok, megfontolások:

- *Ne „aprónként” változzék*, mert akkor nagyon sok lépést kell tenni, sőt valószínűleg nagyot már nem is tudna mozdítani az elemeken egy kb. ugyanakkora lépésközű „újrarendezés”. Pl. felezdjön!
- *Relatív prímelek* legyenek az egymást követő lépésközök, mert ellenkező esetben sok elemet ugyanabba a részsorozatba sorolván, fölöslegesen hasonlítjuk egymáshoz. Pl. csökkenő prímelek!

Ezek után maga az algoritmus, amely alapja a már ismert [javított beillesztéses rendezés](#).

Alg:

```
Eljárás ShellRendezés (Konstans N:Egész, Változó X:THk) :
Változó
  i, j, k, m:Egész
  segéd:TH
```

```

m:=m0 [kezdő lépésköz]
Ciklus amíg m≥1 [lépésköz-ciklus]
  Ciklus k=m+1-től 2*m-ig [k: az m. részsorozat 2. eleme]
    [Javított beillesztéses rendezés
    m-lépésközű sorozatra:]
    Ciklus i=k-től N-ig m-esével
      j:=i-m; segéd:=X(i)
      Ciklus amíg j>0 és X(j)>segéd
        X(j+m):=X(j); j:=-m
      Ciklus vége
      X(j+m):=segéd
    Ciklus vége
  Ciklus vége
  m:=Következő(m)
Ciklus vége
Eljárás vége.
  [10=2k-1 esetén:]
Függvény Következő(Konstans m:Egész):Egész
  [Ef: m=2k-1]
  Következő:=(m+1) Div 2 -1
  [uf: Következő(m)=2k-1-1]
Függvény vége.
  vagy [m0,m1=egymást követő Fibonacci-számok esetén
  m0,m1 globális változók:]
Függvény Következő(Konstans m:Egész):Egész
  [Ef: m,m1=egymást követő Fibonacci-számok]
Változó
  mm,m2:Egész
  m2:=m-m1; mm:=m1; m1:=m2
  Következő:=mm
  [Uf: Következő(m),m1=egymást követő Fibonacci-számok]
Függvény vége.
  [Pl. m0=m=13, m1=8 ⇒ (m,m1)=(8,5),(5,3),(3,2),(2,1)]

```

*Megjegyzés:*

David Russell (1971) „kimérte” a Shell rendezés hatékonyságát:  $100 \leq N \leq 60000$  mellett, különböző lépésköz-függvény esetén így találta:

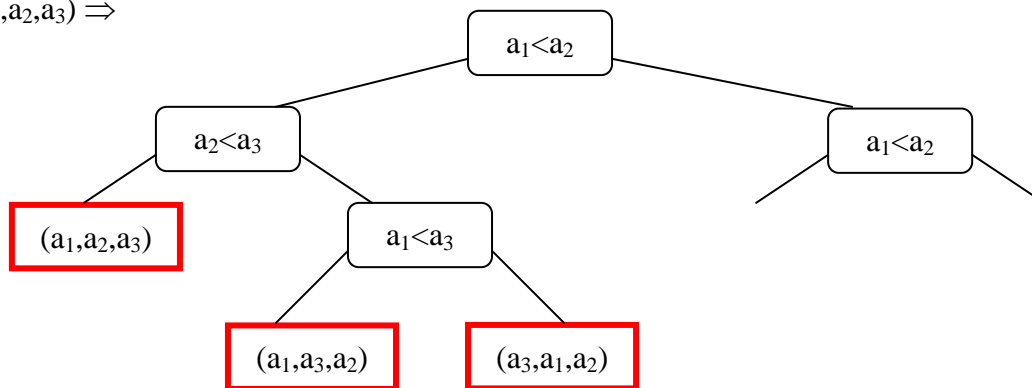
- $m_k = 2^k + 1$  esetén  $1.09 * N^{1.27}$
- $m_k = 2^k - 1$  esetén  $1.22 * N^{1.26}$

## 2. EGY CSÖPP RENDEZÉSELMÉLET

### 2.1. Minimális hasonlításszám

N elem rendezéseinek feleltessünk meg az ún. *döntés (összehasonlítási) v. végrehajtási fát!*

Pl.  $(a_1, a_2, a_3) \Rightarrow$



A fa leveleinek száma 3 elem permutációinak száma:  $3! = 6$ . N elem esetén  $N!$ .

**Állítás:**

A rendezés hasonlításainak száma elemszám függvényében  $H(n) \in \Omega(n \cdot \log(n))$

*Bizonyítás:*

1. Rendeljük az  $n$  elemű sorozathoz a döntési fát!
2. Ekkor a permutációk száma:  $n!$ , ami éppen a leveleinek száma.
3. Legyen  $h := a$  magassága, azaz leghosszabb ágának hossza (vagyis a legkedvezőtlenebb esetben a szükséges hasonlítások száma).
4. Ekkor egy bináris fának legfeljebb (a teljes bináris fának pontosan)  $2^h$  levele van.

Azaz  $n! \leq 2^h$ . Innen  $h \geq \log(n!)$ .

A Stirling-formula legegyszerűbb alakját ( $n! \approx (n/e)^n$ ) felhasználva:

$$H(n) \geq \log(n!) \approx n \cdot \log(n) - n \cdot \log(e) \Rightarrow H(n) \in \Omega(n \cdot \log(n))$$

(A  $H(n) \in \Omega(n \cdot \log(n))$  belátása:

keressük olyan  $c \in \mathbb{R}_+$ -t és  $n_0 \in \mathbb{N}$ , amelyre  $\forall n \geq n_0$ -re teljesül

$$n \cdot \log(n) - n \cdot \log(e) \geq c \cdot n \cdot \log(n)$$

$c$ -re rendezve:

$$1 - \log(e)/\log(n) \geq c$$

Legyen pl.  $c := 0,5$ , akkor  $n_0$  határindexnek megfelelő lesz:

$$1 - \log(e)/\log(n) \geq 0,5 \Leftrightarrow 0,5 \geq \log(e)/\log(n) \Leftrightarrow \log(n) \geq 2 \cdot \log(e) \Leftrightarrow n \geq e^2 = n_0.$$

Q.e.d.

### 2.2. Minimális mozgatósszám

N elem rendezése úgy, hogy a lehető legkevesebb mozgatóst végezze az algoritmus.

Megoldás nyilván a [szétosztó rendezés](#). Hátránya, hogy

1. Megduplázza a helyigényt (sőt).
2. Komoly előzetes elvárással él (Kulcs=rendezés szerint hely indexe).

Tegyük föl (anélkül, hogy magát az algoritmust keressünk), hogy valahonnan rendelkezésre áll a rendezési index-információ<sup>7</sup>, s ezután kell a minimális mozgást megvalósítanunk. Tehát a sorozat megduplázása nélkül, helyben helyretételre kell módszert találnunk.

Lássunk egy számpéldát! Jelölje **S** a sorozat elemeit, az elem indexét az **I**, és a rendezettség szerinti helyének indexét **RI** (**RendIndex**).

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	2	7	1	5	3	6	4
<b>S:</b>	20	70	10	50	30	60	40

A helyrerakás lépései:

1. Helyet csinálunk – az S(1)-ben.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>		7	1	5	3	6	4
<b>S:</b>		70	10	50	30	60	40
	2						
	20						

Ekkor már helyére tehetjük az 1. helyre valót: a 10-et.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	7		5	3	6	4
<b>S:</b>	<b>10</b>	70		50	30	60	40
	2						
	20						

A felszabadult 10 helyére való a 30.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	7	<b>3</b>	5		6	4
<b>S:</b>	<b>10</b>	70	<b>30</b>	50		60	40
	2						
	20						

és így tovább...

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	7	<b>3</b>		<b>5</b>	6	4
<b>S:</b>	<b>10</b>	70	<b>30</b>		<b>50</b>	60	40
	2						
	20						

<sup>7</sup> Láttuk a [számláló rendezés](#)nél, ilyen információ megszerezhető.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	7	<b>3</b>	<b>4</b>	<b>5</b>	6	
<b>S:</b>	<b>10</b>	70	<b>30</b>	<b>40</b>	<b>50</b>	60	
	2						
	20						

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>		<b>3</b>	<b>4</b>	<b>5</b>	6	<b>7</b>
<b>S:</b>	<b>10</b>		<b>30</b>	<b>40</b>	<b>50</b>	60	<b>70</b>
	2						
	20						

Már csak az előzetesen kimentett 20-at kell visszatennünk, és kész.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	6	<b>7</b>
<b>S:</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>	60	<b>70</b>

(Ne is vegyük észre, hogy a 60 meg sem moccan!)

De nézzük az alábbi, picit módosított példát:

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>		3	1	5	7	6	4
<b>S:</b>		30	10	50	70	60	40
	2						
	20						

Ugyanúgy járunk el, ahogy előbb:

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	<b>1</b>	3		5	7	6	4
<b>S:</b>	<b>10</b>	30		50	70	60	40
	2						
	20						

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	1		3	5	7	6	4
<b>S:</b>	10		30	50	70	60	40
	2						
	20						

... és máris visszatehető a 20, sőt egyebet sem tehetünk.

<b>I:</b>	1	2	3	4	5	6	7
<b>RI:</b>	1	2	3	5	7	6	4
<b>S:</b>	10	20	30	50	70	60	40

De baj van: nem vagyunk még készen! :-)

...

A probléma: a permutáció gyakorta (sőt többnyire) –csoportelméleti szóhasználat szerinti– ún. *ciklusok szorzata*, amelyeken belül az elemek egymástól függetlenül permutálhatók a helyükre. A fenti példa ciklusok szorzatára bontva: (132)(754)(6).

Alg:

```

Konstans MaxN:Egész (???)
Típus THk= Tömb(1..MaxN:TH)
           TIndk=Tömb(1..MaxN:0..MaxN) [=0⇒OK; =0⇒i. helyre való]
           TKeresett=Rekord(van:Logikai, hol:Egész)
Eljárás Helyretétel(Konstans N:Egész, Változó8 Ind:TIndk
                    [Változó] S:THk):

  Változó
    j:Egész
    tovább:TKeresett

  tovább.hol:=1; tovább.van:=Igaz
  Ciklus amíg tovább.van
    EgyCiklusHelyre(tovább.hol, Ind, S)
    tovább:=Keresés(Ind, ≠0) [van még ciklus? hol?]
  Ciklus vége
Eljárás vége.
Eljárás EgyCiklusHelyre(Változó j:Egész,
                          Ind:TIndk, S:THk):

  Változó
    r:TH
    i:Egész

```

<sup>8</sup> Bár nem *kimeneti* szerepű az adat!



```

r:=S(j) [S(j) fölszabadítása]
Ind(j):=0 [ez már rendben lesz]
Ciklus amíg Ind(j)≠0
  i:=Kiválasztás(Ind,=j) [hol van a j-vel egyező indexű?]
  S(j):=S(i) [S(i) helyretétele, fölszabadítása]
  Ind(i):=0 [ez már rendben lesz]
  j:=i
Ciklus vége
S(j):=r
Eljárás vége.

```

Hatékonysági elemzés:

$:=_{TH}$	$=_{TH, \leq TH}$	$:=_{\mathbb{N}}$	$=_{\mathbb{N}, \leq \mathbb{N}}$	Indexelés	TH	$\mathbb{N}$
<b>EgyCiklusHelyre</b>						
$2+c_j*1$	0	$(O(N)*c_j+$	$O(N)*c_j$	$O(N)*c_j$	1	1
		<i>Kiválasztás*c<sub>j</sub></i>				
		$+3*c_j)+1$		$+4$		
$2+c_j$	0	$(O(N)+3)*c_j+1$	$O(N)*c_j+1$	$O(N)*c_j+4$	1	1
<b>HelyreTétel (<math>\sum_j c_j = c</math>)</b>						
$\sum_j 2+c_j$	0	$O(N)*c$	$O(N)*c$	$O(N)*c$	0	3
		<i>Keresés*c</i>				
		$+1)$		$+4$		
$O(N)$	0	$\sum_i O(N) +1$	$\sum_i O(N)$	$\sum_i O(N)$	0	3
$O(N)$	0	$O(N)$	$O(N)$	$O(N)$	1	4

## TARTALOM

ProgramozásMódszertan 5. előadás'2004 (vázlat).....	1
1. Rendezési tételek.....	1
1.1. Rendezési tételekről előzetesen.....	1
1.1.1. A feladat .....	1
1.1.2. Fajtái.....	1
1.1.3. Algoritmikus „filozófiák” .....	2
1.1.4. A hatékonyságról.....	2
1.1.5. A specifikációjukról .....	3
1.2. Belső rendezések .....	4
1.2.1. Egyszerű cserés rendezés .....	4
1.2.2. Minimumkiválasztásos rendezés.....	4
1.2.3. Buborékos rendezés.....	5
1.2.4. Javított buborékos rendezés .....	6
1.2.5. Beillesztéses rendezés .....	6
1.2.6. Javított beillesztéses rendezés .....	7
1.3. Speciális rendezések.....	7
1.3.1. Szétosztó rendezés.....	7
1.3.2. Számlálva szétosztó rendezés.....	8
1.3.3. Számláló rendezés .....	9
1.4. Shell rendezés.....	10
2. Egy csöpp rendezéelmélet .....	13
2.1. Minimális hasonlítószám .....	13
2.2. Minimális mozgatószám .....	13