Programozás I.

Rendezések

Sergyán Szabolcs
sergyan.szabolcs@nik.uni-obuda.hu

Óbudai Egyetem
Neumann János Informatikai Kar

2012. november 5.

Rendezési algoritmusok

- A rendezési algoritmusok alapfeladata egy N elemű sorozat nagyság szerinti sorba rendezése.
- Szükséges, hogy a sorozat elemei között értelmezhető legyen a $<$ művelet.
- A rendezéseket általában helyben az eredeti sorozatban hajtjuk végre, így a rendezetlen sorozatot elveszítjük.

$$\boxed{5}\boxed{3}\boxed{9}\boxed{1}\boxed{7} \quad \rightarrow \quad \boxed{1}\boxed{3}\boxed{5}\boxed{7}\boxed{9}$$

- A rendezési algoritmusok bemenete minden esetben egy N elemű X tömb, kimenete pedig ugyanez a tömb, de növekvő módon rendezetten.

Rendezések

1. Egyszerű cserés rendezés
2. Minimumkiválasztásos rendezés
3. Buborékos rendezés
4. Javított buborékos rendezés
5. Beillesztéses rendezés
6. Javított beillesztéses rendezés
7. Rendezés Shell módszerrel

Egyszerű cserés rendezés

Alapötlet

- Hasonlítsuk össze a sorozat első elemét az összes őt követővel.
- Ha egy elem kisebb mint az első, akkor cseréljük meg őket.
- Így a sorozat első helyére a megfelelő (azaz legkisebb) elem kerül.
- Ezt követően ugyanezt tegyük meg a második elemmel, majd az összes következővel.

Egyszerű cserés rendezés

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow 1$**-től** $N-1$**-ig**
    Ciklus $j \leftarrow i+1$**-től** N**-ig**
      Ha $X[i] > X[j]$ **akkor**
        Csere($X[i]$,$X[j]$)
      Elágazás vége
    Ciklus vége
  Ciklus vége
Eljárás vége

Példa

$i=1$	$\rightarrow$	5	3	9	1	7
		3	5	9	1	7
		3	5	9	1	7
		1	5	9	3	7
		1	5	9	3	7
$i=2$	$\rightarrow$	1	5	9	3	7
		1	5	9	3	7
		1	3	9	5	7
		1	3	9	5	7
$i=3$	$\rightarrow$	1	3	9	5	7
		1	3	5	9	7
		1	3	5	9	7
$i=4$	$\rightarrow$	1	3	5	9	7
		1	3	5	7	9

Csere megvalósítása

- Két sorozatbeli elem megcseréléséhez szükséges egy plusz elem a sorozaton kívül, melynek típusa azonos a sorozat elemeinek típusával.

Algoritmus

Eljárás Csere($X[i]$,$X[j]$)
$TEMP \leftarrow X[i]$
$X[i] \leftarrow X[j]$
$X[j] \leftarrow TEMP$
Eljárás vége

Egyszerű cserés rendezés

Hatékonyság

- Helyfoglalás: $N + 1$
- Hasonlítások száma: $\dfrac{N(N-1)}{2}$
- Mozgatások száma: Legalább 0, legfeljebb $3 \cdot \dfrac{N(N-1)}{2}$

Rendezések összehasonlítása

30 elemű sorozatokat hasonlítottunk össze

- A majdnem rendezett sorozatban két elem nem volt a helyén.
- A fordítva rendezett sorozat csökkenő sorrendben volt rendezett.
- A véletlen sorozat az első 30 pozitív egész szám egy véletlen permutációja volt.

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693

Minimumkiválasztásos rendezés

Alapötlet

- Az *Egyszerű cserés rendezés* hibája, hogy általában túl sok cserét hajt végre annak érdekében, hogy az elemek a megfelelő helyre kerüljenek.
- Ezen javíthatunk, ha az aktuális elemet a mögötte állók minimumával cseréljük fel, ha egyáltalán kell cserélni.

Minimumkiválasztásos rendezés

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow 1$**-től** $N-1$**-ig**
    $MIN \leftarrow i$
    Ciklus $j \leftarrow i+1$**-től** N**-ig**
      Ha $X[MIN] > X[j]$ **akkor**
        $MIN \leftarrow j$
      Elágazás vége
    Ciklus vége
    Csere($X[i]$,$X[MIN]$)
  Ciklus vége
Eljárás vége

Példa

$i=1$	$\rightarrow$	5	3	9	1	7
$i=2$	$\rightarrow$	1	3	9	5	7
$i=3$	$\rightarrow$	1	3	9	5	7
$i=4$	$\rightarrow$	1	3	5	9	7
		1	3	5	7	9

Hatékonyság

- Helyfoglalás: $N+1$
- Hasonlítások száma: $\dfrac{N(N-1)}{2}$
- Mozgatások száma: $3 \cdot (N-1)$

- A mozgatások száma akár több is lehet, mint az *Egyszerű cserés rendezésnél*.
- Ennek oka, hogy a külső ciklusban mindenképp cserélünk, annak érdekében, hogy ne kelljen mindig egy összehasonlítást is elvégezni.

Rendezések összehasonlítása

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693
Minimumkiválasztásos	435	29	87	435	29	87	435	29	87

Rendezések

1. Egyszerű cserés rendezés
2. Minimumkiválasztásos rendezés
3. **Buborékos rendezés**
4. Javított buborékos rendezés
5. Beillesztéses rendezés
6. Javított beillesztéses rendezés
7. Rendezés Shell módszerrel

Buborékos rendezés

Alapötlet

- Összehasonlítjuk egymással a szomszédos elemeket, s ha a sorrendjük nem jó, akkor cseréljük meg őket.
- A szomszédok cseréje miatt először a legnagyobb elem fog a helyére kerülni, majd ezt követően a második legnagyobb, és így tovább.
- Az algoritmusban az elemek úgy kerülnek a sorozatbeli helyükre a legnagyobbtól kezdve, mint a felfelé szálló buborékok.

Buborékos rendezés

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow N$**-től** 2**-ig** -1**-esével**
    Ciklus $j \leftarrow 1$**-től** $i-1$**-ig**
      Ha $X[j] > X[j+1]$ **akkor**
        Csere($X[j]$,$X[j+1]$)
      Elágazás vége
    Ciklus vége
  Ciklus vége
Eljárás vége

Példa

$i=5$	$\rightarrow$	5	3	9	1	7
		3	5	9	1	7
		3	5	9	1	7
		3	5	1	9	7
		3	5	1	7	9
$i=4$	$\rightarrow$	3	5	1	7	9
		3	5	1	7	9
		3	1	5	7	9
		3	1	5	7	9
$i=3$	$\rightarrow$	3	1	5	7	9
		1	3	5	7	9
		1	3	5	7	9
$i=2$	$\rightarrow$	1	3	5	7	9
		1	3	5	7	9

Hatékonyság

- Helyfoglalás: $N+1$
- Hasonlítások száma: $\dfrac{N(N-1)}{2}$
- Mozgatások száma: Legalább 0, legfeljebb $3 \cdot \dfrac{N(N-1)}{2}$

Rendezések összehasonlítása

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693
Minimumkiválasztásos	435	29	87	435	29	87	435	29	87
Buborékos	435	29	87	435	435	1305	435	231	693

Rendezések

Alapötlet

- Ha a *Buborékos rendezés* belső ciklusában egyáltalán nincs már csere, akkor az algoritmust kár folytatni.
- Ha volt csere, de az utolsó csere például a sorozat közepénél volt, akkor az utolsó csere helyétől a sorozat végéig a sorozat már rendezett, tehát kár azzal a résszel foglalkozni.

Algoritmus

Eljárás Rendezés(N,X)
    $i \leftarrow N$
    Ciklus amíg $i \geq 2$
        $CS \leftarrow 0$
        Ciklus $j \leftarrow$ **1-től** $i - 1$**-ig**
            Ha $X[j] > X[j+1]$ **akkor**
                Csere($X[j]$,$X[j+1]$)
                $CS \leftarrow j$
            Elágazás vége
        Ciklus vége
        $i \leftarrow CS$
    Ciklus vége
Eljárás vége

Példa

$i = 5$	$\rightarrow$	5	3	9	1	7
		3	5	9	1	7
		3	5	9	1	7
		3	5	1	9	7
		3	5	1	7	9
$i = 4$	$\rightarrow$	3	5	1	7	9
		3	5	1	7	9
		3	1	5	7	9
		3	1	5	7	9
$i = 2$	$\rightarrow$	3	1	5	7	9
		1	3	5	7	9

Javított buborékos rendezés

- Helyfoglalás: $N+1$
- Hasonlítások száma: Legalább $N-1$, legfeljebb $\dfrac{N(N-1)}{2}$
- Mozgatások száma: Legalább 0, legfeljebb $3 \cdot \dfrac{N(N-1)}{2}$

- A *Buborékos rendezés*hez képest szignifikáns javulás az átlagos végrehajtási időben van.

Rendezések összehasonlítása

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693
Minimumkiválasztásos	435	29	87	435	29	87	435	29	87
Buborékos	435	29	87	435	435	1305	435	231	693
Javított buborékos	254	29	87	435	435	1305	429	231	693

Rendezések

1. Egyszerű cserés rendezés
2. Minimumkiválasztásos rendezés
3. Buborékos rendezés
4. Javított buborékos rendezés
5. **Beillesztéses rendezés**
6. Javított beillesztéses rendezés
7. Rendezés Shell módszerrel

Alapötlet

- Egy elem önmagában rendezett.
- A második elemet tegyük a helyére.
- A harmadikat és az összes következőt tegyük a helyére.

Beillesztéses rendezés

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow 2$**-től** N**-ig**
    $j \leftarrow i - 1$
    Ciklus amíg $j > 0$ **és** $X[j] > X[j+1]$
      Csere($X[j]$,$X[j+1]$)
      $j \leftarrow j - 1$
    Ciklus vége
  Ciklus vége
Eljárás vége

Példa

$i = 2$	$\rightarrow$	5	3	9	1	7
		3	5	9	1	7
$i = 3$	$\rightarrow$	3	5	9	1	7
$i = 4$	$\rightarrow$	3	5	9	1	7
		3	5	1	9	7
		3	1	5	9	7
		1	3	5	9	7
$i = 5$	$\rightarrow$	1	3	5	9	7
		1	3	5	7	9

Hatékonyság

- Helyfoglalás: $N + 1$
- Hasonlítások száma: Legalább $N - 1$, legfeljebb $\dfrac{N(N-1)}{2}$
- Mozgatások száma: Legalább 0, legfeljebb $3 \cdot \dfrac{N(N-1)}{2}$

Rendezések összehasonlítása

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693
Minimumkiválasztásos	435	29	87	435	29	87	435	29	87
Buborékos	435	29	87	435	435	1305	435	231	693
Javított buborékos	254	29	87	435	435	1305	429	231	693
Beillesztéses rendezés	58	29	87	435	435	1305	260	231	693

Rendezések

Javított beillesztéses rendezés

Alapötlet

- A *Beillesztéses rendezés*ben túl sok cserét hajtunk végre annak érdekében, hogy egy elem a helyére kerüljön.
- Hasznosabb lenne, ha a szükséges elemeket hátrább tolnánk eggyel.

Javított beillesztéses rendezés

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow 2$**-től** N**-ig**
    $j \leftarrow i - 1$
    $Y \leftarrow X[i]$
    Ciklus amíg $j > 0$ **és** $X[j] > Y$
      $X[j+1] \leftarrow X[j]$
      $j \leftarrow j - 1$
    Ciklus vége
    $X[j+1] \leftarrow Y$
  Ciklus vége
Eljárás vége

Példa

$i = 2$	$\rightarrow$	5	3	9	1	7
		5	5	9	1	7
$i = 3$	$\rightarrow$	3	5	9	1	7
$i = 4$	$\rightarrow$	3	5	9	1	7
		3	5	9	9	7
		3	5	5	9	7
		3	3	5	9	7
$i = 5$	$\rightarrow$	1	3	5	9	7
		1	3	5	9	9
		1	3	5	7	9

Javított beillesztéses rendezés

Hatékonyság

- Helyfoglalás: $N + 1$
- Hasonlítások száma: Legalább $N - 1$, legfeljebb $\dfrac{N(N-1)}{2}$
- Mozgatások száma: Legalább $2 \cdot (N-1)$, legfeljebb $2 \cdot (N-1) + \dfrac{N(N-1)}{2}$

Rendezések összehasonlítása

	Majdnem rendezett sorozat			Fordítva rendezett sorozat			Véletlen sorozat		
	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.	Hason.	Csere	Mozg.
Egyszerű cserés	435	29	87	435	435	1305	435	231	693
Minimumkiválasztásos	435	29	87	435	29	87	435	29	87
Buborékos	435	29	87	435	435	1305	435	231	693
Javított buborékos	254	29	87	435	435	1305	429	231	693
Beillesztéses rendezés	58	29	87	435	435	1305	260	231	693
Javított beillesztéses	58		87	435		493	260		289

Rendezések

1. Egyszerű cserés rendezés
2. Minimumkiválasztásos rendezés
3. Buborékos rendezés
4. Javított buborékos rendezés
5. Beillesztéses rendezés
6. Javított beillesztéses rendezés
7. Rendezés Shell módszerrel

Rendezés Shell módszerrel

Alapötlet

- Hasonlítsuk először össze az egymástól távoli elemeket, és ha kell cseréljük meg őket
- Így az egyes elemek gyorsan közel kerülnek a végleges helyükhöz

Rendezés Shell módszerrel

Algoritmus

Eljárás Rendezés(N,X)
  $L \leftarrow L_0$
  Ciklus amíg $L \geq 1$
    Ciklus $K \leftarrow L+1$**-tól** $2L$**-ig**
      Ciklus $i \leftarrow K$**-tól** N**-ig** L**-esével**
        $j \leftarrow i - L$
        $Y \leftarrow X[i]$
        Ciklus amíg $j > 0$ **és** $X[j] > Y$
          $X[j+L] \leftarrow X[j]$
          $j \leftarrow j - L$
        Ciklus vége
        $X[j+L] \leftarrow Y$
      Ciklus vége
    Ciklus vége
    $L \leftarrow$Következő(L)
  Ciklus vége
Eljárás vége

Algoritmus

Eljárás Rendezés(N,X)
  Ciklus $i \leftarrow 2$**-től** N**-ig**
    $j \leftarrow i - 1$
    $Y \leftarrow X[i]$
    Ciklus amíg $j > 0$ **és** $X[j] > Y$
      $X[j+1] \leftarrow X[j]$
      $j \leftarrow j - 1$
    Ciklus vége
    $X[j+1] \leftarrow Y$
  Ciklus vége
Eljárás vége

Rendezés Shell módszerrel

Módosított algoritmus

Eljárás Rendezés(N,X)
  $L \leftarrow L_0$
  Ciklus amíg $L \geq 1$
    Ciklus $i \leftarrow L+1$**-től** N**-ig**
      $j \leftarrow i - L$
      $Y \leftarrow X[i]$
      Ciklus amíg $j > 0$ **és** $X[j] > Y$
        $X[j+L] \leftarrow X[j]$
        $j \leftarrow j - L$
      Ciklus vége
      $X[j+L] \leftarrow Y$
    Ciklus vége
    $L \leftarrow$Következő(L)
  Ciklus vége
Eljárás vége

Rendezés Shell módszerrel

- L_0 nem lépheti túl N-t
- A Következő(L) függvény meghatározására pár példa:

$2^k - 1$ alakú eset

Eljárás Következő(L)
    Következő$\leftarrow (L + 1)/2 - 1$
Eljárás vége

Fibonacci szám alakú eset

Eljárás Következő(L)
    $L2 \leftarrow L - L1$
    $L \leftarrow L1$
    $L1 \leftarrow L2$
    Következő$\leftarrow L1$
Eljárás vége

L meghatározására további javaslatok

Shell sorozat

$$L \leftarrow \left\lfloor \frac{N}{2^k} \right\rfloor,$$

ahol $k = 1, 2, \ldots$
Ebben az esetben a rendezés $O(n^2)$-es

Pratt sorozat

L értékei a $2^p \cdot 3^q$ alakban előállítható számok, azaz pl.

$$\{1, 2, 3, 4, 6, 8, 9, 12 \ldots\}$$

Ebben az esetben a rendezés $O(n \cdot \log^2 n)$-es

Knuth sorozat

L értékei a $(3^k - 1)/2$ értékei, azaz pl.

$$\{1, 3, 13, 40, 121, \ldots\}$$

Ebben az esetben a rendezés $O(n^{3/2})$-es

Felhasznált irodalom

- Szlávi Péter, Zsakó László: Módszeres programozás: Programozási tételek (Mikrológia 19). ELTE TTK, 2002