



Algoritmusok és adatstruktúrák 7. előadás

Pusztai Pál
pusztai@sze.hu

Tartalom

- Egyszerű rendezések
 - Buborékredezés
 - Rendezés kiválasztással
 - Rendezés beszúrással
- Keresések
 - Lineáris keresés
 - Rendezetlen, ill. rendezett adatok között
 - Bináris keresés
- Indextáblás rendezések és keresések



Rendezés és keresés

- **Feladat:** Rendezzünk egy tömbben lévő adatsort növekvő (nem csökkenő) sorrendbe!

- **Típus**

ELEM Egész

/* A rendezendő elemek típusa */

TOMB Egydimenziós ELEM tömb

/* Az elemeket tároló tömb típusa */

- **Műveletigény**

- Egyszerű rendezések: n^2 nagyságrendű.
- Hatékony rendezések: $n \cdot \log_2 n$ nagyságrendű.



Buborékrendezés

K	1.	2.	3.	4.	5.
3	1	1	1	1	1
6	3	2	2	2	2
2	6	3	3	3	3
1	2	6	4	4	4
5	4	4	6	5	5
4	5	5	5	6	6

Buborékrendezés



Buborékrendezés

Funkció	Azonosító	Típus	Jelleg
A rendezendő elemek	A	TOMB	I, M, O
Az elemek száma	N	Egész	I
Két elem cseréjéhez	CS	ELEM	M
A helyére kerülő elem indexe	I	Egész	M
A cserélgető ciklus változója	J	Egész	M

Buborékrendezés

```
/* Buborékrendezés */  
BUBREND(A,N)  
for I  $\leftarrow$  1,N-1  
    /* I. elemet a helyére */  
    for J  $\leftarrow$  N,I+1,-1  
        if A[J]<A[J-1]  
            /* A J. és J-1. elemek cseréje */  
            CS  $\leftarrow$  A[J]  
            A[J]  $\leftarrow$  A[J-1]  
            A[J-1]  $\leftarrow$  CS
```



Rendezés kiválasztással

K	1.	2.	3.	4.	5.
3	1	1	1	1	1
6	6	2	2	2	2
2	2	6	3	3	3
1	3	3	6	4	4
5	5	5	5	5	5
4	4	4	4	6	6

Rendezés kiválasztással

Rendezés kiválasztással

Funkció	Azonosító	Típus	Jelleg
A rendezendő elemek	A	TOMB	I, M, O
Az elemek száma	N	Egész	I
Két elem cseréjéhez	CS	ELEM	M
A helyére kerülő elem indexe	I	Egész	M
Az aktuális minimum indexe	K	Egész	M
A minimumkereső ciklus változója	J	Egész	M

Rendezés kiválasztással

```
/* Rendezés kiválasztással */  
KIVALREND(A,N)  
for I  $\leftarrow$  1,N-1  
    /* I. elemet a helyére */  
    K  $\leftarrow$  I  
    for J  $\leftarrow$  I+1,N  
        if A[J]<A[K]  
            K  $\leftarrow$  J  
  
    if K>I  
        /* Az I. és K. elemek cseréje */  
        CS  $\leftarrow$  A[I]  
        A[I]  $\leftarrow$  A[K]  
        A[K]  $\leftarrow$  CS
```



Rendezés beszúrással

K	1.	2.	3.	4.	5.
3	3	2	1	1	1
6	6	3	2	2	2
2	2	6	3	3	3
1	1	1	6	5	4
5	5	5	5	6	5
4	4	4	4	4	6

Rendezés beszúrással

Rendezés beszúrással

Funkció	Azonosító	Típus	Jelleg
A rendezendő elemek	A	TOMB	I, M, O
Az elemek száma	N	Egész	I
Két elem cseréjéhez	CS	ELEM	M
A helyére kerülő elem	X	ELEM	M
A helyére kerülő elem indexe	I	Egész	M
A helykészítő ciklus változója	J	Egész	M

Rendezés beszúrással

```
/* Rendezés beszúrással */  
BESZURREND(A,N)  
for I  $\leftarrow$  2,N  
    /* I. elem beszúrása az előtte lévő rendezett részbe */  
    X  $\leftarrow$  A[I]  
    /* Helykészítés hátraléptetéssel */  
    J  $\leftarrow$  I-1  
    while (J $\geq$ 1) AND (A[J] $>$ X)  
        A[J+1]  $\leftarrow$  A[J]  
        J  $\leftarrow$  J-1  
    /* I. elemet a helyére */  
    A[J+1]  $\leftarrow$  X
```



Lineáris keresések

- **Feladat:** Keressünk meg egy adott elemet egy rendezetlen adatsorban!



Lineáris keresések

- **Feladat:** Keressünk meg egy adott elemet egy rendezetlen adatsorban!

Funkció	Azonosító	Típus	Jelleg
Az elemek	A	TOMB	I
Az elemek száma	N	Egész	I
A keresendő elem	X	ELEM	I
A keresés eredménye	HOL	Egész	O
A keresés sikeressége	VAN	Logikai	O
A keresőciklus változója	I	Egész	M

Lineáris keresések

/* Lineáris keresés rendezetlen adatok között */

LINKER(A,N,X,HOL)

$I \leftarrow 1$

while ($I \leq N$) AND ($A[I] \neq X$)

$I \leftarrow I + 1$

$VAN \leftarrow I \leq N$

if VAN

$HOL \leftarrow I$

return VAN

Lineáris keresések

```
/* Lineáris keresés rendezett adatok között */  
LINKERREND(A,N,X,HOL)  
I ← 1  
while (I ≤ N) AND (A[I] < X)  
    I ← I + 1  
VAN ← (I ≤ N) AND (A[I] = X)  
HOL ← I  
return VAN
```

Bináris keresés

- **Feladat:** Keressünk meg egy adott elemet egy rendezett adatsorban!

Bináris keresés

- **Feladat:** Keressünk meg egy adott elemet egy rendezett adatsorban!

Funkció	Azonosító	Típus	Jelleg
Az elemek	A	TOMB	I
Az elemek száma	N	Egész	I
A keresendő elem	X	ELEM	I
A keresés eredménye	HOL	Egész	O
A keresés sikeressége	VAN	Logikai	O
A keresés helyének kezdőindexe	I	Egész	M
A keresés helyének végindexe	J	Egész	M
A keresés helyének középső indexe	K	Egész	M

Bináris keresés

2	2	3	4	4	6
I		K			J
<hr/>					
2	2	3	4	4	6
			I	K	J

A bináris keresés lépései a 4 keresésekor

Bináris keresés

	2	2	3	4	4	6
	I		K			J
<hr/>						
	2	2	3	4	4	6
	I	J				
	K					
<hr/>						
	2	2	3	4	4	6
J	I					

A bináris keresés lépései az 1 keresésekor

Bináris keresés

```
/* Bináris keresés */  
BINKER(A,N,X,HOL)  
I ← 1  
J ← N  
VAN ← hamis  
while (I ≤ J) AND NOT VAN  
    K ← (I+J) DIV 2  
    if A[K]=X  
        VAN ← igaz  
    else if X < A[K]  
        J ← K-1  
    else  
        I ← K+1  
if VAN  
    HOL ← K  
else  
    HOL ← I  
return VAN
```



Bináris keresés

- Megjegyzés:
 - Ha a keresett elem nem található meg az adatok között, akkor a HOL azt mutatja, hogy hol lenne a helye a rendezettség szerint.

Indextáblás rendezések

Név szerinti indextábla	Sorszám	Adatok Név	Telefonszám	Tel. szám sz-i indextábla
5	1	Szabó	222222	3
3	2	Varga	777777	1
7	3	Kovács	111111	5
1	4	Takács	444444	4
4	5	Halász	333333	7
6	6	Vadász	666666	6
2	7	Madarász	555555	2



Indextáblás rendezések

- **Feladat:** Rendezzük egy tömbben lévő adatsort növekvő (nem csökkenő) sorrendbe a kiválasztásos rendezési módszerrel, indextábla használatával!
- **Típus**
INDEXTABLA Egydimenziós egész tömb /* Az indextábla */
- **Megoldás:** Tetszőleges rendezési algoritmus átírható indextáblássá az alábbiak szerint:
 - Az indextáblát fel kell tölteni az eredeti sorrendet kifejező indexekkel.
 - Az adatokra mindig közvetve, az indextáblán keresztül kell hivatkozni.
 - Az adatok helyett az indexeik mozognak, tehát az elemek cseréjét átírjuk az indexeik cseréjére.

Indextáblás rendezések

Funkció	Azonosító	Típus	Jelleg
A rendezendő elemek	A	TOMB	I
Az indextábla	IT	INDEXTABLA	M, O
Az elemek száma	N	Egész	I
Két index cseréjéhez	CS	Egész	M
A helyére kerülő elem indexe	I	Egész	M
Az aktuális minimum indexe	K	Egész	M
A minimumkereső ciklus változója	J	Egész	M

Indextáblás rendezések

```
/* Rendezés kiválasztással, indextáblával */  
KIVALRENDIT(A,IT,N)  
/* Az indextábla feltöltése */  
for I  $\leftarrow$  1,N  
    IT[I]  $\leftarrow$  I  
for I  $\leftarrow$  1,N-1  
    /* I. elemet a helyére */  
    K  $\leftarrow$  I  
    for J  $\leftarrow$  I+1,N  
        if A[IT[J]] < A[IT[K]]  
            K  $\leftarrow$  J  
  
    if K > I  
        /* Az I. és K. indexek cseréje */  
        CS  $\leftarrow$  IT[I]  
        IT[I]  $\leftarrow$  IT[K]  
        IT[K]  $\leftarrow$  CS
```



Indextáblás keresések

- **Feladat:** Keressünk meg egy adott elemet egy indextáblával rendezett adatsorban!



Indextáblás keresések

- **Feladat:** Keressünk meg egy adott elemet egy indextáblával rendezett adatsorban!

Funkció	Azonosító	Típus	Jelleg
Az elemek	A	TOMB	I
Az indextábla	IT	INDEXTABLA	I
Az elemek száma	N	Egész	I
A keresendő elem	X	ELEM	I
A keresés eredménye	HOL	Egész	O
A keresés sikeressége	VAN	Logikai	O
A keresés helyének kezdőindexe	I	Egész	M
A keresés helyének végindexe	J	Egész	M
A keresés helyének középső indexe	K	Egész	M

Indextáblás keresések

/* Bináris keresés indextáblával */

BINKERIT(A,IT,N,X,HOL)

$I \leftarrow 1$

$J \leftarrow N$

$VAN \leftarrow \text{hamis}$

while ($I \leq J$) AND NOT VAN

$K \leftarrow (I+J) \text{ DIV } 2$

if $A[IT[K]] = X$

$VAN \leftarrow \text{igaz}$

else if $X < A[IT[K]]$

$J \leftarrow K - 1$

else

$I \leftarrow K + 1$

if VAN

$HOL \leftarrow K$

else

$HOL \leftarrow I$

return VAN

Indextáblás keresések

- Megjegyzés:
 - A HOL a keresett elem indextáblabeli helyét mutatja.

