

jövő héten nincs  
előadás

ismétlés:

$LSz(n)$  = hány elemi lépést tesz az algoritmus az " $n$ "  
mireltű inputon a legrosszabb esetben

Nem mindig tudjuk, hogy mi a legrosszabb eset,  
eset van, hogy az első bőlőt adunk a  
lépésszáma.

$LSz(n)$  jellegére  $n \geq n_0$  küszöbindex fölött  
ragyunk kiváncsiak:

pl.:  $LSz(n) \sim n$   
 $LSz(n) \sim n^2$   
 $LSz(n) \sim n^3 \dots$

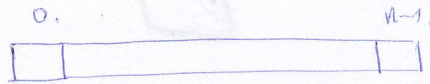
Def.:  $LSz(n) = O(f(n))$  ha  $\exists c, n_0: LSz(n) \leq c \cdot f(n)$  ha  $n \geq n_0$

példa:  $LSz(n) \leq 8n^2 + 3n - 7 \leq 8n^2 + 3n \stackrel{n \geq 1}{\leq} 8n^2 + 3n^2 = 11n^2$

példák:  $(n)$   $[100n^2 + \log_2 n + 7]$   $n^3/10000$   $(7n)$   
 $(38n + 17)$   $(10n - 3)$   $[n^2 - n + 5]$   $[3n^2 - 27n + 3]$



### Minimumkeresés:



$min = T[0]$

$min\_hely = 0$

for  $i = 1$  to  $n-1$ :

if  $T[i] < min$ :

$min = T[i]$

$min\_hely = i$

### (SELECTION SORT)

Kiválasztásos rendezés: Adatok növekvő sorrendbe

Típus: mind különböző értéket tartalmaz

0. minimumkeresés  $i = 0$  to  $n-1$  elemek között  
és a min-t  $T[0]$ -ba mozgatom

1. minimumkeresés  $i = 1$  to  $n-1$  elemek között  
és ott a min-t  $T[1]$ -be mozgatom

⋮

### PSEUDOKÓD:

for  $j = 0$  to  $n-2$

$min = T[j]$

$min\_hely = j$

for  $i = j+1$  to  $n-1$

if  $T[i] < min$

$min = T[i]$

$min\_hely = i$

cseré( $T[j], T[min\_hely]$ )

2 - 2 -



pl.:

|   |   |    |   |    |
|---|---|----|---|----|
| 8 | 5 | -4 | 0 | 10 |
|---|---|----|---|----|

$j=0$  után

|    |   |   |   |    |
|----|---|---|---|----|
| -4 | 5 | 8 | 0 | 10 |
|----|---|---|---|----|

Helyesség:

a tömb rendezett lesz az algoritmus végén:

mert: az első körben meglett a legkisebb  
és az kerül az első helyre ...

(formálisan teljes indukcióval lehetne  
bizonyítani)

Lépésenként:

összehasonlítást és értékelést  
tekintünk elemi lépésnek

$j$  index mentén  $n$  ciklus fut le,  
amelyek mindegyikében legfeljebb  $c \cdot n$   
~~lépést~~ lépést teszünk

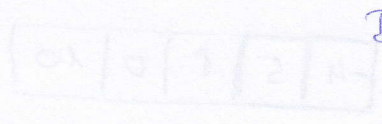
$$LS_2(n) \leq n \cdot c \cdot n = c \cdot n^2$$

$$LS_2(n) = O(n^2)$$





A: 21 .... 36 -5  
16 .... 31



B: 13 ..., 20, 29, -36 -5  
8 ..., 15, 24 ... 31

C: 9...12, 17...20, 25...28, 33...36  
4...7, 12...15, 20...23, 28...31

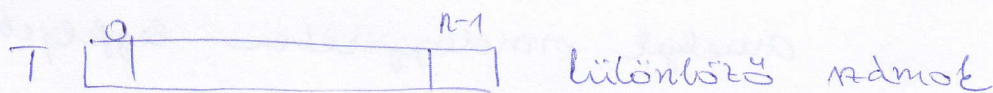
D: 7 8 11 12 15 16 18 20 23 24 27 28 31 32 35 36  
23 6 7 10 11 14 15 18 19 22 23 26 27 30 31

E:

1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

Ötlet meg is van!

pl.: Keresési feladat:



keresni a értéket!

ha nem tudunk semmit a tömbről.

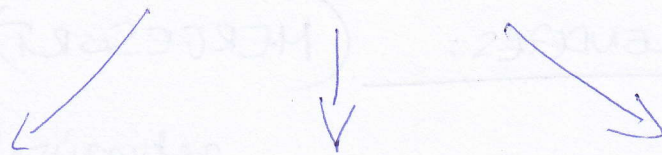
lineáris keresés : egészeivel vizsgáljuk a tömb  
elemeket  
 $LS(n) \leq n$

ha T tömb ismeretlen!

lineáris keresés

ötlet: hasonlítani össze s-et és T[közép]-elemet





$$T[\text{közép}] < s$$



keresés a 2. felében

$$T[\text{közép}] = s$$



megvan

$$T[\text{közép}] > s$$



keresés az 1. felében

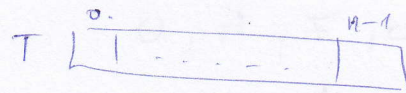
$$\text{közép} : \left\lfloor \frac{n+1}{2} \right\rfloor$$

$$2^{\lfloor \log_2 n \rfloor}$$

hány összehasonlítás van?  
 lépésmódul:  $O(\log_2(n))$

(Ha nincs találat,  
 akkor megvan, hogy  
 hol lenne)

### BESZÁRAKÁSOS RENDSZER:



$T[0]$  rendszert

$T[1]$ -et ellenőrző tömbbe

$T[2]$  ellenőrző tömbbe

benyitás  $\rightarrow$  keresés  
 +  
 mozgató

$LSZ(n)$ :

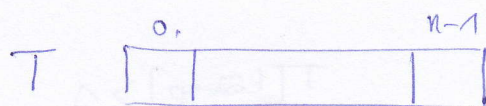
n darab benyitás  $\times$   $\left( \overset{\substack{\text{rendszert tömbben} \\ \text{bináris keresés}}}{O(\log n)} + \overset{\text{mozgató}}{O(n)} \right)$

$$= O(n \cdot \log n) + O(n^2) = O(n^2)$$

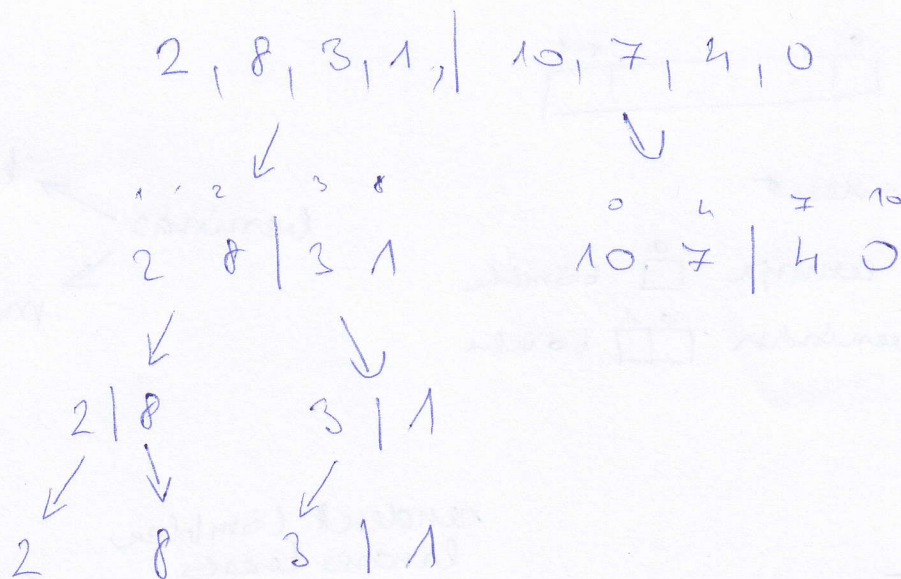
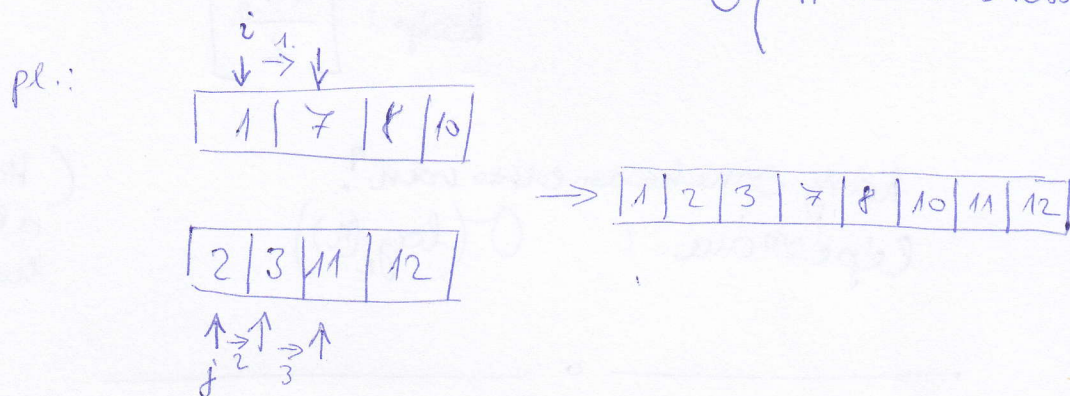


# ÖSSZEFESZÍTÉSES RENDELÉS: (MERGESORT)

rekurzív!



- 1.) rendezni T első felét  $\rightarrow$  A tömb összefeszítéses rendezés
- 2.) rendezni T második felét  $\rightarrow$  B tömb összefeszítéses rendezés
- 3.) A-t és B-t összefészelem  $\rightarrow O(A \text{ hossza} + B \text{ hossza})$



LSz:  $O(n \cdot \log n)$