

13. tétel

Adatszerkezetek és algoritmusok vizsga

Frissült: 2013. január 28.

Gyorsrendezés (Quicksort)

Egy hatékony rendezési algoritmus. Az „Oszd meg és uralkodj!” elvén működik, két fázisa van:

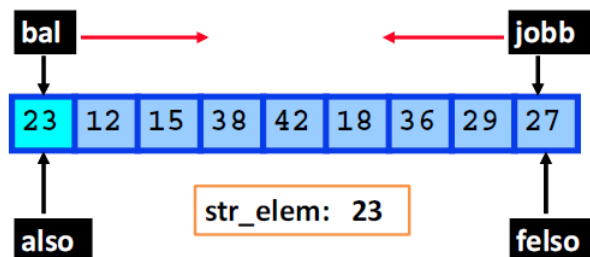
- Partíciós fázis - Megosztás

A partícióban egy ún. strázsát (pivot) kell választani. Ettől balra minden elemnek kisebbnek, jobbra nagyobbaknak kell majd lennie.

< strázsza	strázsza	> strázsza
------------	-----------------	------------

Megosztás menete:

- adott egy also, felső, bal jobb mutató.
- Bármely elem jó strázsának, pl. a bal szélső elem.
- beállítjuk a bal és a jobb végét
- a jelzéseket addig mozgatjuk míg szembetalálkoznak.
 - * A bal elemet jobbra míg a strázsaelemnél $\leq$ elemre mutat
 - * a jobb elemet balra, míg a strázsaelemnél $\geq$ elemre mutat

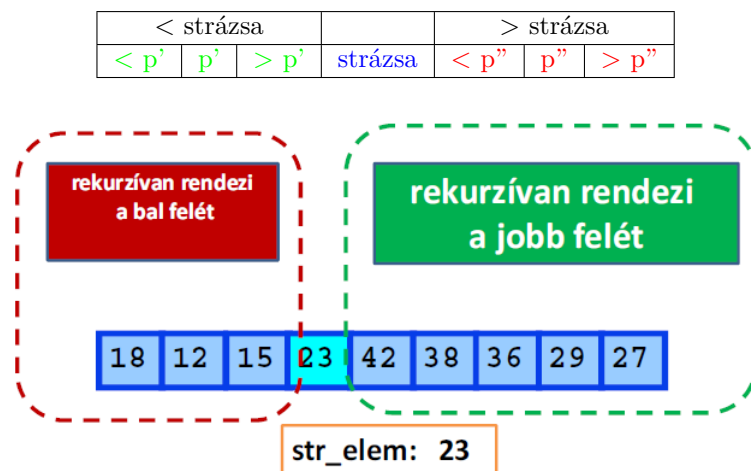


- miután megállunk, látjuk, hogy a bal és a jobb rossz sorrendben vannak, kicseréljük a bal és a jobb értékét.
- továbbmegyünk, mikor szembetalálkoznak, leállunk.
- aztán kicseréljük a strázsát a jobb elemmel. Új strázsza a jobb elem.

Sebesség: megvizsgál minden elemet egyszer $O(n)$

- Rendezési fázis

Ugyanezt ezt a bal és jobb félre is alkalmazni kell, rekurzívan.

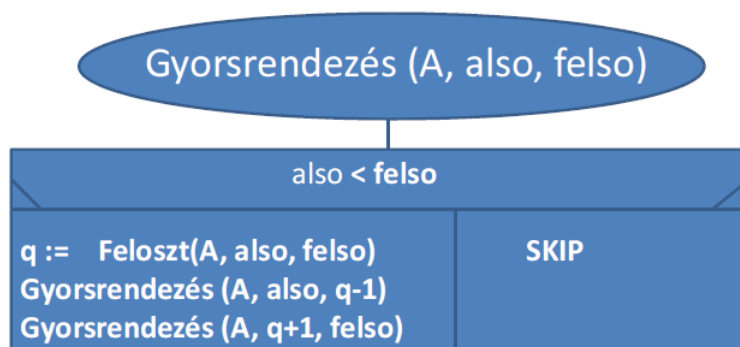


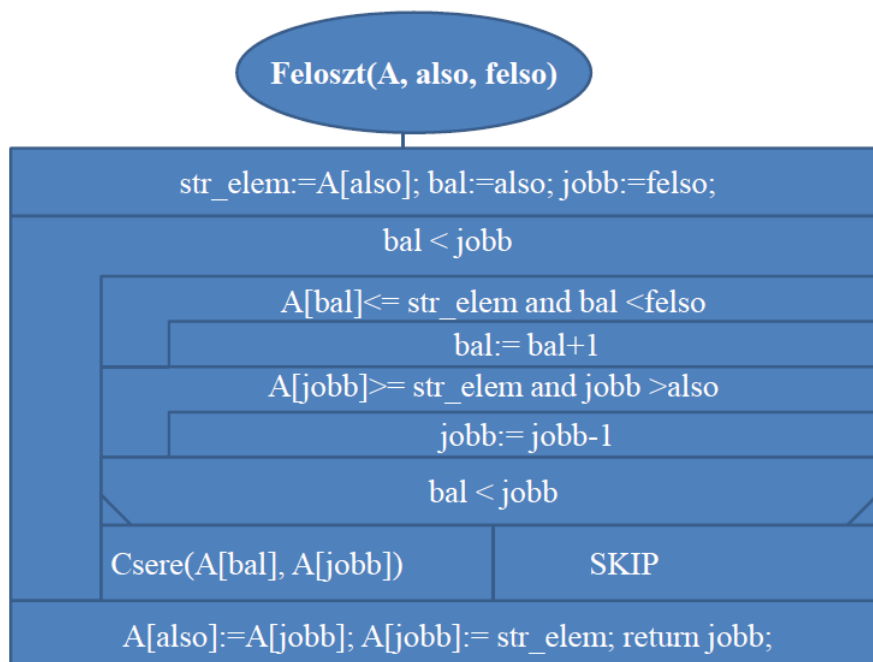
Sebesség: adatok kétfelé osztása $O(\log_2 n)$

Implementáció

```
quicksort( void *a, int also, int felso )
{
    int pivot;
    /* Terminálási feltétel! */
    if ( felso > also )
    {
        pivot = feloszt( a, also, felso ); Oszd meg
        quicksort( a, also, pivot-1 );      Uralkodj
        quicksort( a, pivot+1, felso );    ..
    }
}
```

Struktogram





Műveletigénye

Jó esetben:

$$O(n) \cdot O(\log_2 n) = O(n \log n)$$

DE, ha az adatok részben vagy teljesen rendezettek, akkor

- Minden partíció létrehoz egy 0 méretű feladatot és egy $n-1$ méretűt!
- Partíciók száma: minden n időigénye $O(n)$, összesen $nO(n)$ vagyis

$$O(n^2)$$

is lehet! Tehát nem garantált a sebesség, az a pivot jó megválasztásától függ!

Nem csak bal szélső strázsát lehetne választani, lehetne középsőt, utolsót is. (Részben) rendezettnél a középső jó lehet :

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Esetleg véletlen pivotot választunk.