12. Gyors rendezés
(Quick Sort)

Az egy elemet helyre rakó rendezések osztályába tartozik. A rendezés lényege, hogy tetszés szerint kiválasztunk egy elemet, kivesszük a sorozatból, majd a sorozat többi tagját a következőképpen helyezzük el a kivett elem két oldalán:

- egyik oldalra a kivett elemnél nem nagyobbakat
- másik oldalra a kivett elemnél nagyobbakat.

Az így létrejött sorozat mindkét oldalán ismét végrehajtjuk a rendezést, amíg teljesen rendezett sorozatot nem kapunk.

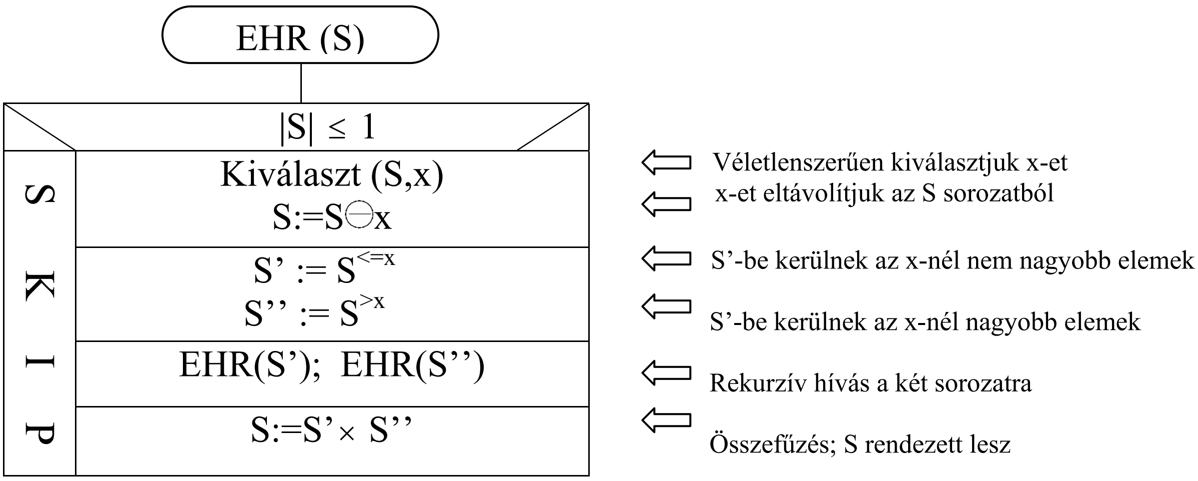
S rendezett $\Leftrightarrow \exists x$:

$S = S' \times S''$

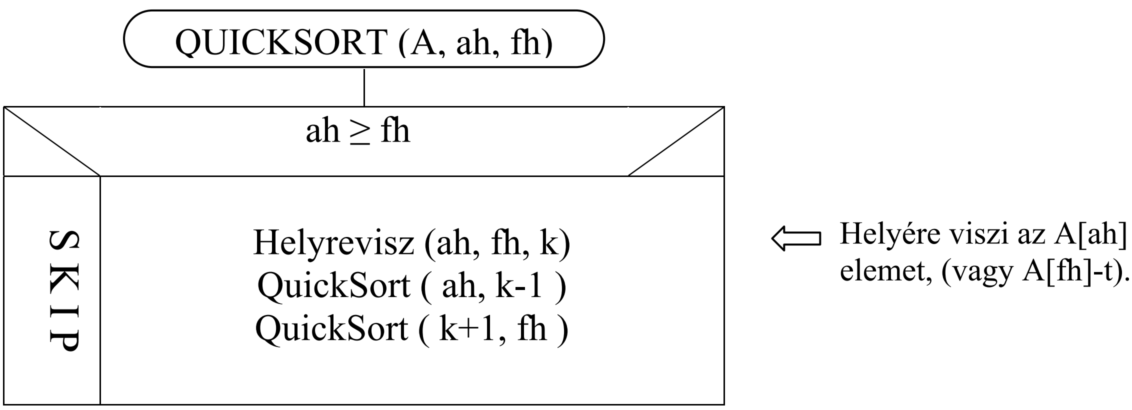
S' elemei $\leq$ x $\leq$ S'' elemei

S' és S'' rendezettek

Az eljárás általános megfogalmazása az S sorozatra:



A quicksort eljárást az A[1..n] tömbre szokták megfogalmazni; a véletlen elemkiválasztás nem szerepel benne, mert a tömb általában tekinthető véletlenszerűnek.



A helyrevisz eljárás két változatát ismertetjük.
(az 1. változat A[ah]-t viszi a helyére, a 2. változat A[fh]-t viszi a helyére.)

<u>Helyrevisz 1. változat</u>

Az A tömb A[u..v] részletében helyére viszi az A[u] elemet. Meghíváskor $u<v$, ugyanis az $u \geq v$ esetben a QuickSort SKIP ága hajtódik végre.

Ötlet:

i j
u

5	2	5	7	3	7	4	4	5

i→ ←j
csere

- i-vel (n+1)-ről elindulva, jobbra haladva lépjük át azokat az elemeket, amelyekre $A[i] \leq A[u]$.
  j-vel v-ről indulva, balra haladva lépjük át azokat az elemeket, amelyekre $A[u] \leq A[j]$.

- Az ábrán egy olyan megállás látható, amelyben $i<j$, és természetesen $A[i] > A[u]$ és $A[j] < A[u]$. Ekkor cseréljük meg a két elemet.
  Lépjünk tovább mindkét mutatóval, és folytassuk ugyanígy tovább.

A csere után:

i j

5	2	5	4	3	7	4	7	5

i→ ←j

i tovább lép, j azonban nem. Csere!

Újabb csere után:

i és j átlépték egymást, természetesen nincs csere.

5	2	5	4	3	4	7	7	5

↑ ↑
i j

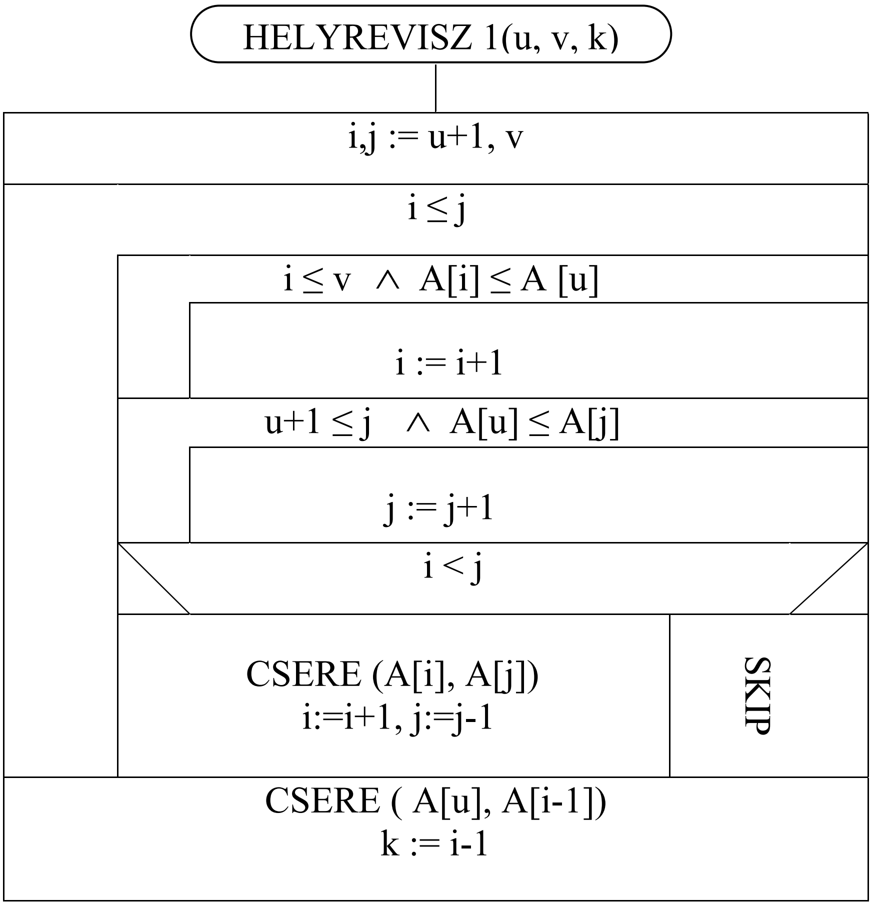
- Az eljárás mindig így áll meg, hogy i és j átlépik egymást: $j < i$. Azért van ez így, mert az A[u]-nak egyenlő elemeket mindketten átlépik. ⇒ A fő ciklus feltétele: $i \leq j$

- Az i jobbra léptetésére „fizikai korlát”: $i \leq v$
- A j balra léptetésére „fizikai korlát” : $u+1 \leq j$

} Ezeket beépítjük a két függvénybe.

- Ha $j < i$, vagyis i és j átlépik egymást, akkor A[u]-t megcseréljük A[i-1]-gyel, és $k=i-1$ mutatja az A[u] elem helyét.

- Ha speciálisan $i=u+1$, azaz nincs A[u]-nál kisebb-egyenlő elem, akkor a csere hatása: $A[u] \leftrightarrow A[u]$, ami helyes (csak fölösleges).



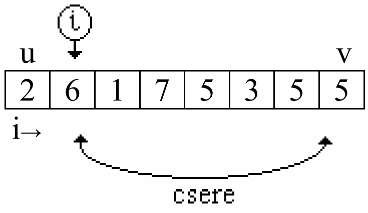
$MÖ_{H1}(n) = n-1$, sőt ez az összehasonlítási szám fix módon n-1 $\Rightarrow Ö_{H1}(n) = n-1$

$MCS_{H1}(n) = \left\lfloor \frac{n-1}{2} \right\rfloor + 1$

$mCS_{H1}(n) = 1$

Helyrevisz 2. változat

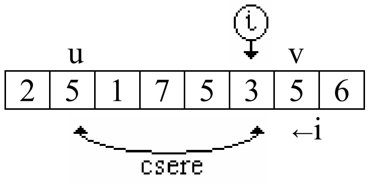
Q: u < v változatlanul, de most az A[v]-t visszük helyre.
Ötlet:



- i-vel u-ról indulva, jobbra haladva lépjük át az $A[i] \leq A[v]$ elemeket.

- Legyen a továbblépés feltétele $i \leq v \;\wedge\; A[i] \leq A[v] \;\Rightarrow\; i := i+1$.
  (Lehetne i<v is, de most maradjunk a fentinél)

- Ha i<v és A[i] > A[v], akkor cseréljük meg a két elemet:



  Csere (A[i], A[v]).
  Állítsuk be u-t és v-t a keresés új szűkebb tartománya határaira:
  u := i; v:= v-1

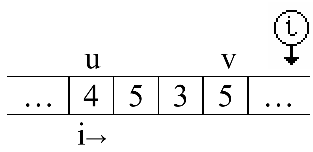
- Váltsunk irányt és i-vel v-ről indulva, balra haladva lépjünk át az $A[i] \geq A[u]$ elemeket.

- A továbblépés feltétele: $u \leq i \;\wedge\; A[u] \leq A[i] \;\Rightarrow\; i := i-1$

- Ha u<i és A[u] > A[i], akkor cseréljük meg a két elemet, állítsuk be u-t és v-t:
  Csere (A[u], A[i])
  u := u+1; v := i
  és ismét váltsunk irányt….

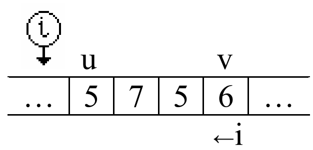
- Hogyan áll le az algoritmus?

  Jobbra haladva i átlépi v-t, pl.:



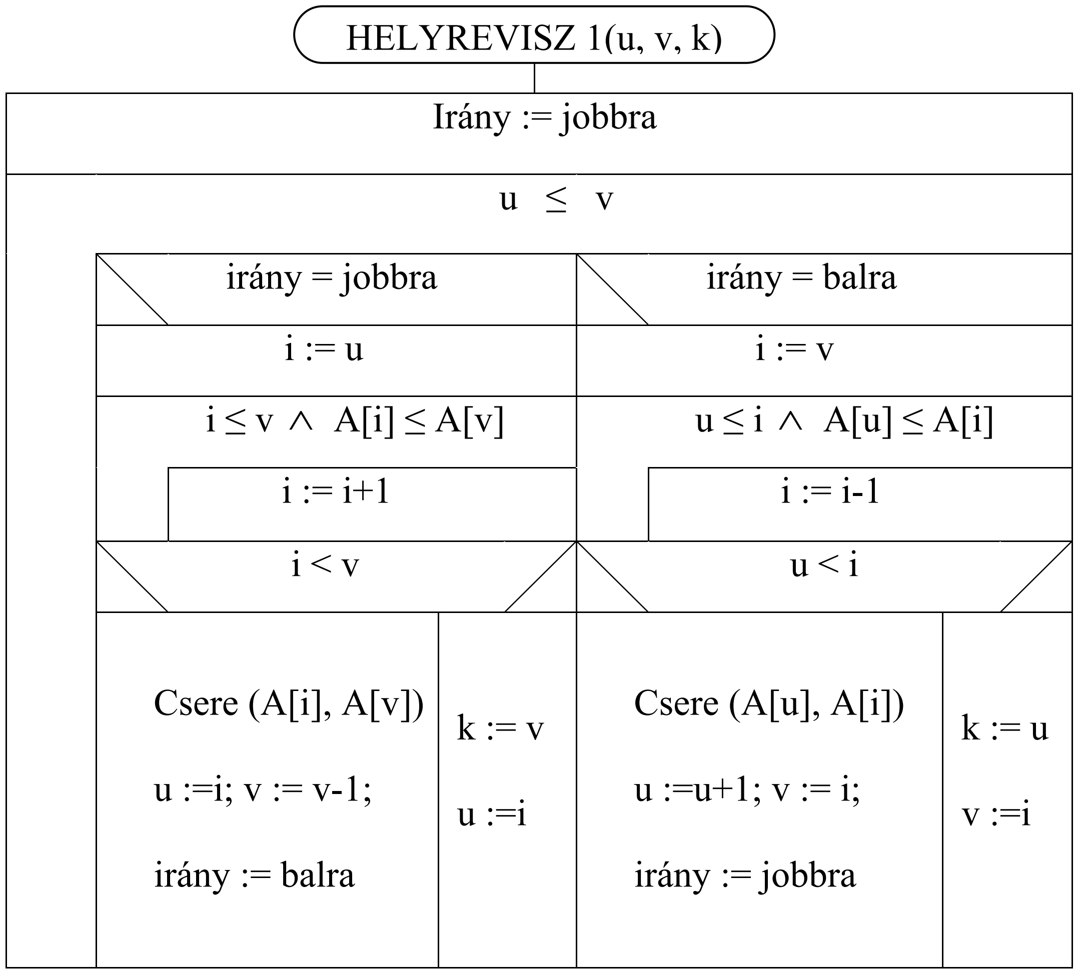
  Ekkor A[v] a helyén van.
  Legyen k := i-1 (vagy k := v) és u := i.

  Balra haladva i átlépi u-t, pl. .:



  Ekkor A[u] a helyén van.
  Legyen k := u és v := i.

Végül tehát u és v átlépték egymást.



Hatékonyság elemzés

Még egyszer az algoritmus: Hívás:
QuickSort(A, 1, 1)

QuickSort(A, ah, fh)

	ah $\geq f(n)$
SKIP	Helyrevisz (ah, fh, k) QuickSort (A, ah, k-1) QuickSort (A, k+1, fh)

A Helyrevisz utasítás összehasonlítás száma:
a (rész)tömb elemeinek száma-1.

$$Ö_{Helyre}(n) = n-1$$

, ahol n $\geq$ 2, hiszen 1-elemű vagy üres résztömbre nem hívjuk meg.

A QuickSort teljes összehasonlítás száma erősen függ az input permutációitól.

Legrosszabb eset: az abszolút kiegyensúlyozatlan szétvágás. Ekkor a helyrevitt elem mindig a vizsgált résztömb valamelyik szélére kerül.
Ez következik be, például: triviálisan az 1 2 … n vagy az n(n-1) … 2 1 esetben.

$$\begin{cases} MÖ_{Quick}(n) = (n-1) + MÖ_{Quick}(n-1) \\ \\ MÖ_{Quick}(2) = 1 \end{cases}$$

Kifejtve:

$$MÖ_{Quick}(n) = (n-1) + (n-2) + ... + 2 + 1 = \frac{n(n-1)}{2} = \Theta(n^2)$$

A QuickSort eljárás nem illeszkedik az előrendezettséghez, például, rendezett input sorozat esetén négyzetesen működik.

Legjobb eset: az abszolút kiegyensúlyozott szétvágás. Ekkor a helyrevitt elem mindig középre kerül és a résztömb feleződik, kb. két egyenlő részre válik szét.

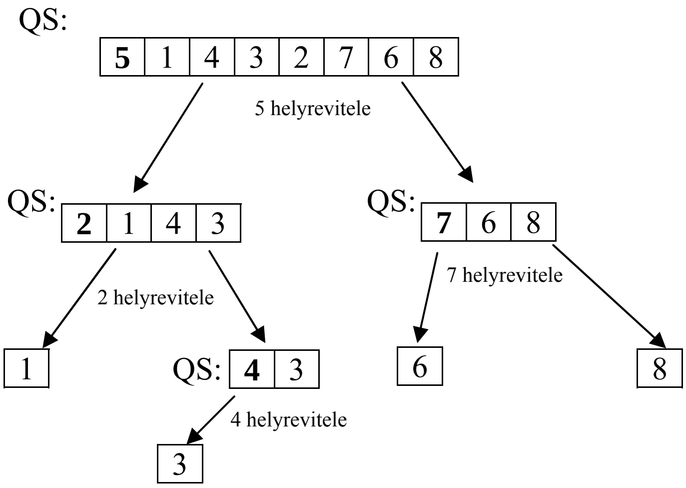
Ha például QuickSorttal rendezzük a következő tömböt:

5	1	4	3	2	7	6	8

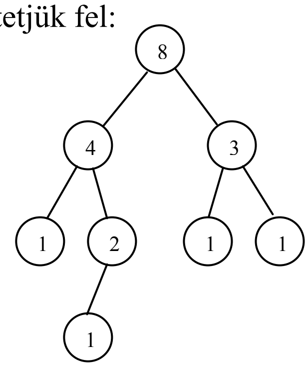
Akkor az 5 helyrevitele (I. eljárás) után a következő szétvágás következik be:

2	1	4	3	**5**	7	6	8

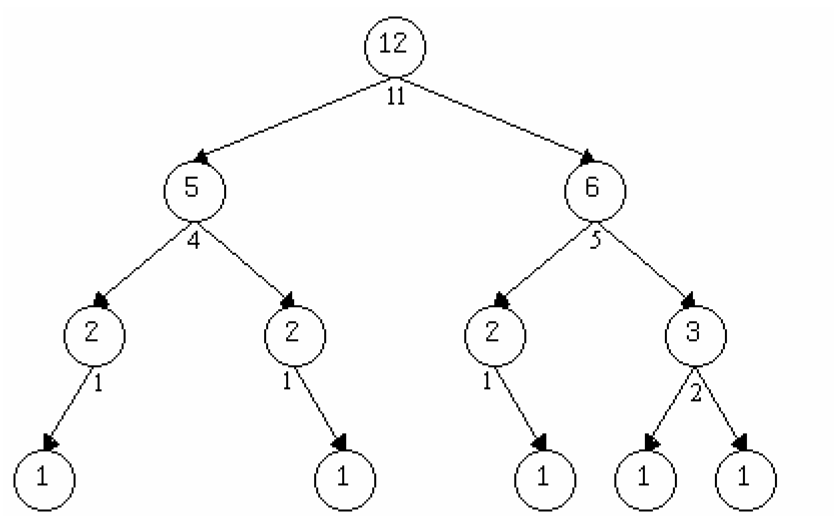
Folytatva a 4-, illetve a 3-elemű résztömbök rendezését, végül a következő rekurzív hívási fához jutunk:



Ha csak a rendezendő (rész)tömb elemszámát tüntetjük fel:



Egy másik, fiktív példa konkrét adatok nélkül n=12 elemre a következő:



A rekurzív hívási fa magassága:

k=3=$\lfloor \log_2 n \rfloor$.

Általában is igaz, hogy abszolút kiegyensúlyozott szétvágás esetén:

k=$\lfloor \log_2 n \rfloor$.

Az összehasonlítások száma a Helyrevisz eljárásból adódik, amelyet a legalsó szint kivételével, az afeletti szinten meghívunk.
Erre a fára példa:

Ö(12)=11+4+5+1+1+1+2=25.

Egyszerű becslés: $mÖ_{Quick}(n)$-re:

Azon szintek száma, amelyeken a Helyrevisz eljárást meghívjuk: $\lfloor \log_2 n \rfloor$.

Az összehasonlítások számának összege minden szinten: $\le (n-1)$.

Így

$$mÖ_{Quick}(n) \le (n-1)\,\lfloor \log_2 n \rfloor < \log_2 n\,.$$

Ugyanezt megkaphatjuk kissé „szabályosabb” módon úgy, hogy felírjuk a minimális összehasonlítás szám rekurzív függvényét.

Jelölje $f(n) := mÖ_{Quick}(n)$

$$
\begin{cases}
f(0) = f(1) = 0 \\[2ex]
f(n) = (n-1) + f\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right) + f\left(\left\lceil \frac{n-1}{2} \right\rceil\right) & (n \geq 2)
\end{cases}
$$

Ezután pedig teljes indukcióval bebizonyítjuk, hogy $f(n) < n \log_2 n$.

<u>Átlagos eset:</u> $AÖ_{Quick}(n) = ?$

Jelöljük az egyszerűség kedvéért: $f(n) := AÖ_{Quick}(n)$

Tegyük fel, hogy minden input permutáció egyformán valószínű.

Ekkor ez a valószínűség: $\frac{1}{n}$!

Ha A[1] a rendezés szerint k-adik a sorban, akkor a Helyrevisz eljárás végrehajtása után az A[k] helyre kerül és egy (k-1) és egy (n-k) hosszú résztömböt kell tovább rendezni QuickSorttal.

Jelöljük ezt az esetet (n|k)-val. Az átlagos összehasonlítás számot nyilván úgy kapjuk meg, ha a k=1, 2, …, n eseteket mind figyelembe vesszük $\frac{1}{n}$ valószínűséggel: $f(n) = \frac{1}{n}\sum_{k=1}^{n} f(n \mid k)$.

Egy k értékre pedig: $f(n \mid k) = n - 1 + f(k+1) + f(n-k)$.

Tehát $f(n) = \frac{1}{n}\sum_{k=1}^{n}[(n-1) + f(k-1) + f(n-k)]$

$$
\begin{aligned}
f(n) &= (n-1) + \frac{1}{n}\sum_{k=1}^{n}[f(k-1) + f(n-k)] = \\
&= (n-1) + \frac{1}{n}[\underbrace{f(0)}_{0} + f(n-1) + f(1) + f(n-2) + \ldots + f(n-1) + \underbrace{f(0)}_{0}] \\
f(n) &= (n-1) + \frac{2}{n}\sum_{k=1}^{n} f(k)
\end{aligned}
$$

Azt <u>állítjuk</u>, hogy ennek a rekurzív egyenletnek a megoldására

$$f(n) \leq 2n \ln n \qquad (n \geq 1).$$

Bizonyítás: $n = 1$ $\quad f(1) = 0$ $\quad$ ezt tudjuk, és valóban $2 \ln 1 = 0$

$n = 2$ (nem kellene ugyan megnézni, de …) $\quad f(2) = 1 \leq 2 \ln 2 = 2 * 0{,}69 = 1{,}38$

tegyük fel, hogy igaz az összefüggés 1, 2, …, $(n-1)$-re, ekkor

$$f(n) = (n-1) + \frac{2}{n}\sum_{k=1}^{n-1} f(k) \le (n-1) + \frac{2}{n}\sum_{k=1}^{n-1} 2k\ln k =$$

$$= (n-1) + \frac{4}{n}\sum_{k=2}^{n-1} k\ln k \le (n-1) + \frac{4}{n}\int_2^n x\ln x\,dx =$$

$$= (n-1) + \frac{4}{n}\left(\left[\frac{x^2\ln x}{2} - \frac{x^2}{4}\right]_2^n\right) = (n-1) + \frac{4}{n}\frac{n^2}{4}\left[2\ln n - 1\right] - \frac{4}{n}\left(\underbrace{2\ln 2 - 1}_{0,3863}\right) \le$$

$$\le 2n\ln n + (n-1) - n - \frac{4}{n}0,39 < 2n\ln n$$

Azt kaptuk, hogy $AÖ_{Quick}(n) < 2n\ln n \approx 1,3863 n\log_2 n$.