15. tétel

Adatszerkezetek és algoritmusok vizsga

Frissült: 2013. január 30.

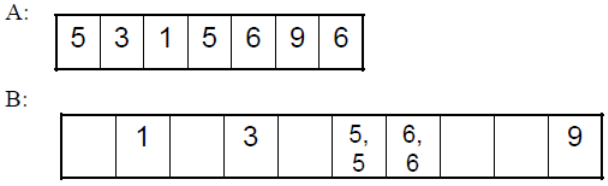
Edényrendezés

Tegyük fel, hogy tudjuk, hogy a bemenő elemek ($A[1..n]$ elemei) egy m elemű U halmazból kerülnek ki, pl. "$\forall A[i]$-re igaz, hogy $A[i] \in [1..m]$.

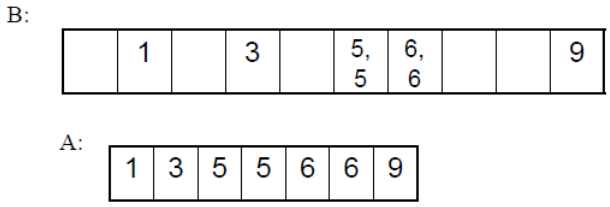
Lefoglalunk egy U elemeivel indexelt B tömböt (m db ládát), először mind üres. B segédtömb elemei lehetnek pl. láncolt listák.

Lépések:

1. végigolvassuk az A-t, és az $s = A[i]$ elemet a $B[s]$ lista végére fűzzük.



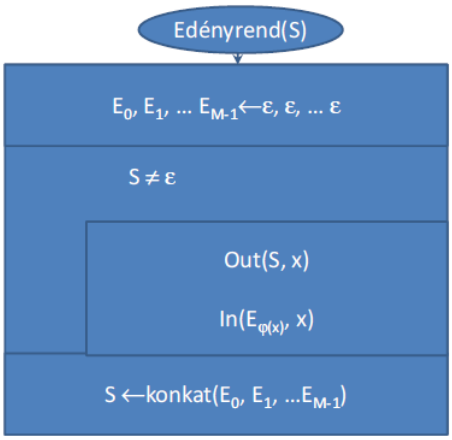
2. elejétől a végéig növő sorrendben végigmegyünk B-n, és a $B[i]$ listák tartalmát visszaírjuk A-ba.



Általánosabb megfogalmazás. Legyen K az S sorozat elemeinek típusértékhalmaza, $\phi : K \to [0..M-1]$ olyan függvény, amire igaz, hogy ha $j(k_1) < j(k_2)$, akkor $k_1 < k_2$.

Legyenek $E_0, E_1, \ldots E_{M-1}$ edények, melyek éppen olyan sorozatok, mint S. Az egyes edényekben megmarad az S-beli elemek ottani relatív sorrendje.

Struktogram.



Ahhoz, hogy Edényrend(S) egyszeri szétrakással rendezzen, elégséges, ha:

- minden edényben legfeljebb egy elem van, ($\forall i \in [0, M-1] : |E_i| \leq 1$) vagy:
- az egyes edényekben csak azonos elemek vannak, vagy:
- az egyes edények rendezettek.

Ha az előző feltételek valamelyike teljesül, akkor tökéletes az edényrendezés.

Ennek műveletigénye: $\theta(n)$, ahol $n = |S|$.

Példa: nagyon sok embert kell magasság szerint sorba rakni $\Longrightarrow$ megéri minden egyes testmagasság számára egy edényt létrehozni.

Lépésszám

- B létrehozása: $O(m)$
- első fázis: $O(n)$
- második fázis: $O(n+m)$ összesen: $O(n+m)$

Ez gyorsabb, mint az általános alsó korlát, ha pl. $m \leq c * n$!!

Radix rendezés

Tegyük fel, hogy a kulcsok összetettek, több komponensből állnak, $(t_1, ...t_k)$ alakú szavak, ahol a t_i komponens az L_i rendezett típusból való, a rendezés lexikografikus.

Példa: Legyen $(U; <)$ a huszadik századi dátumok összessége az időrendnek megfelelő rendezéssel:

$$L_1 = \{1900; 1901; ...; 1999\} \qquad n_1 = 100$$
$$L_2 = \{\text{január}, \text{február}, ..., \text{december}\} \qquad n_2 = 12$$
$$L_3 = \{1; 2; ...; 31\} \qquad n_3 = 31$$

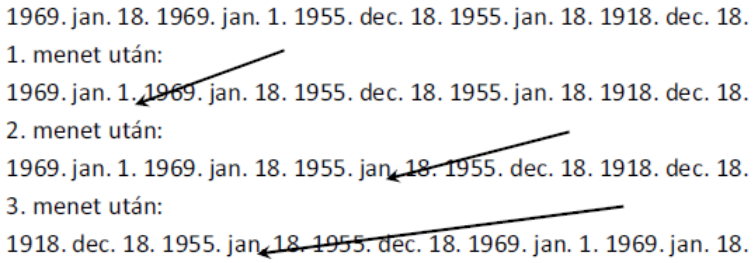
A dátumok rendezése éppen az L_i típusokból származó lexikografikus rendezés lesz.

Menete.

- Rendezzük a sorozatot az utolsó, a k-adik komponensek szerint edényrendezéssel!
- A kapott eredményt rendezzük a k-1-edik komponensek szerint edényrendezéssel
- stb.

Fontos, hogy az edényrendezésnél az elemeket a ládában mindig a lista végére tettük. Így, ha két azonos kulcsú elem közül az egyik megelőzi a másikat, akkor a rendezés után sem változik a sorrendjük. Ez egy *konzervatív* rendezés.

Példa.



Általános leírás. Az $e = e_d e_{d-1} ... e_2 e_1$ számot jobbról balra, az alacsony helyiértékek felől indulva pozíciónként szétrakja edényekbe, majd összefűzi az edények tartalmát.

Definíció. S „i-rendezett” (jelölés: $x \leq_i y$), ha minden

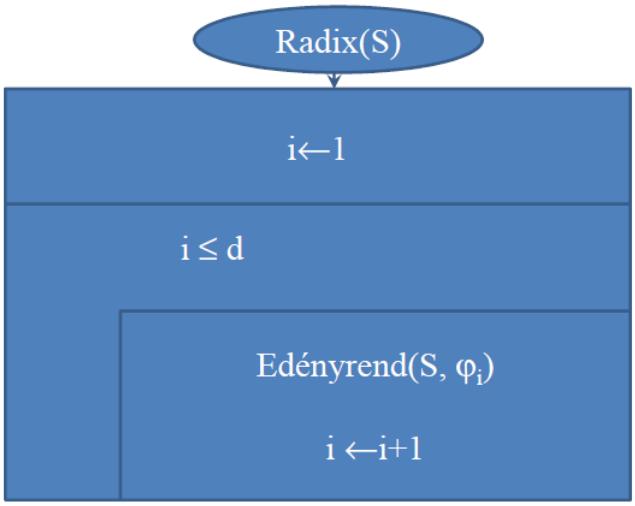
- $x = x_d x_{d-1} ... x_2 x_1$ és $y = y_d y_{d-1} ... y_2 y_1$ –ra: $x <_0 y$
- $x \leq_i y \iff x_i < y_i$ vagy $x_i = y_i$ és $\leq_{i-1} y (i > 0)$
- Ekkor a „d-rendezés” a közönséges rendezés

Az i. pozíción a φ_i függvényt alkalmazzuk: $\varphi_i(e) = e_i$ Az i. pozíción végrehajtott szétrakás és összefűzés után S "i-rendezett" lesz.

Hatékonyság

$2 * d$-szer megyünk végig az S sorozaton, így $T(n) = \theta(d * |S|)$

Struktogram



Szokásos implementáció

S fejelemes láncolt lista

Az edényeket egy „fej” és egy „vége” mutató ábrázolja.

A szétrakás és az összefűzés az elemek láncolásával megoldható. Összefűzéskor nem kell az egyes edények részlistáit végigolvasni, hanem egy darabban lehet őket láncolni.

Leszámláló rendezés

Tegyük fel, hogy van n db bemeneti elem, s ezek mindegyike 1 és k közötti egész szám.

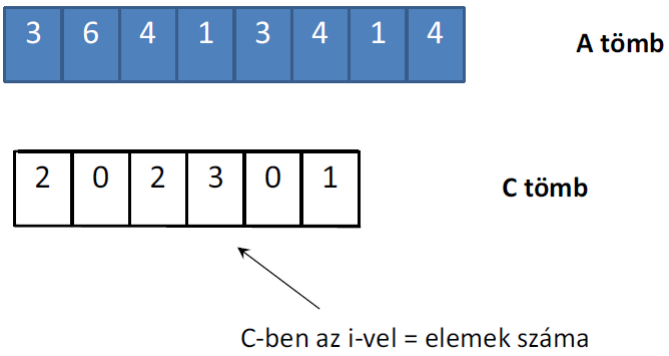
Az alapötlet: meghatározzuk minden egyes x bemeneti elemre azoknak az elemeknek a számát, amelyek kisebbek, mint az x.

Ezután x-et közvetlenül a saját pozíciójára tudom helyezni.

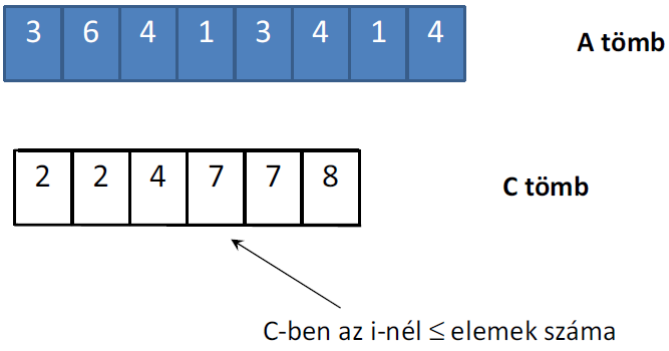
Legyen a bemenet az $A[1..n]$ tömb, a kimenet a $B[1..n]$ tömb. Mindkettő hossza: $hossz[A] = hossz[B] = n$ Szükség van még egy $C[1..k]$ tömbre átmeneti munkaterületként.

Menete

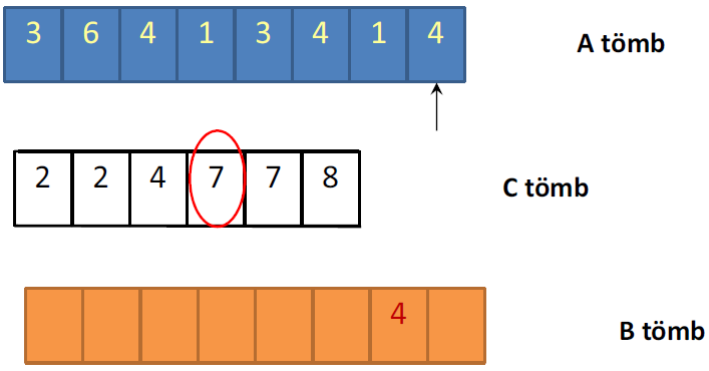
1. Végigmegyünk az A-n, és ha egy elem értéke i, akkor megnöveljük C[i] értékét eggyel.



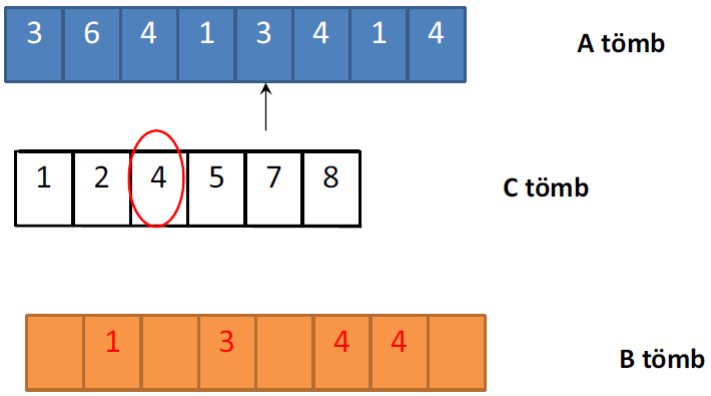
2. Minden i-re 1..k között meghatározzuk, hogy hány olyan bemeneti elem van, amelyiknek az értéke $\leq$ i (összegzés C-n)



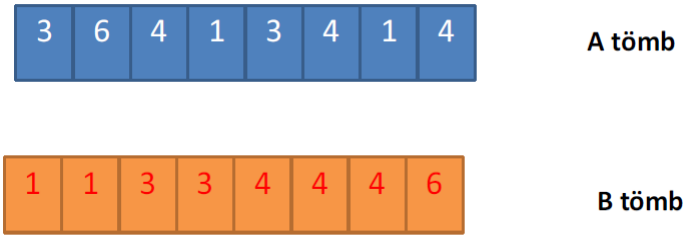
3. Minden j-re n..1 között A[j]-t betesszük B megfelelő pozíciójába - ezt a C-ből állapítjuk meg.



...



... ... a végén:



Ha betettük, akkor a C[A[j]] értékét csökkentjük, így a következő vele egyenlő elem már elé kerül, vagyis így stabil lesz a rendezés, az egyenlő elemeknél megtartja az eredeti sorrendet.

Pszeudokód

```
for i ← 1 to k do
   C[i] ← 0
for j ← 1 to hossz(A) do
   C[A[j]] ← C[A[j]] + 1          C-ben az i-vel = elemek száma
for i ← 2 to k do
   C[i] ← C [i] + C[i-1]
for j ← hossz(A) downto 1 do      C-ben az i-vel ≤ elemek száma
   B[C[A[j]]] ← A[j]
   C[A[j]] ← C[A[j]] - 1
```