A számítástudomány alapjai

Katona Gyula Y.

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Rendezések

Rendezés (Sorting)

Feladat

Adott az $(U, <)$ rendezett halmaz véges $S = \{s_1, s_2, \ldots, s_{n-1}, s_n\}$ részhalmaza.
Összehasonlításokkal rendezzük az S elemeit a rendezés szerint növekvő sorrenbe, azaz keressünk olyan σ permutációt, hogy $s_{\sigma(1)} < s_{\sigma(2)} < \cdots < s_{\sigma(n)}$.

Input: tömb, láncolt lista, (vagy bármi)
Output: általában, mint az Input
Lépések: elemek mozgatása, cseréje, összehasonlítása

A rendezés önmagában is előforduló feladat, de előjön, mint hasznos adatstruktúra is. Rendezett halmazban könnyebb keresni (pl. telefonkönyv).

- Hány összehasonlítás kell a legrosszabb esetben?
- Hány összehasonlítás kell átlagos esetben?
- Hány csere kell a legrosszabb esetben?
- Mennyi plusz tárhely szükséges?

Buborék-rendezés (Bubble sort)

Input: $A[1:n]$ (rendezetlen) tömb
Ha valamely i-re $A[i] > A[i+1]$, akkor a két cella tartalmát kicseréljük. A tömb elejéről indulva a cserélgetve eljutunk a tömb végéig. Ekkor a legnagyobb elem $A[n]$-ben van. Ismételjük ezt az $A[1:n-1]$ tömbre, majd az $A[1:n-2]$ tömbre, stb.

procedure buborék
(az $A[1:n]$ tömböt növekvően (nem csökkenően) rendezi *)*
for $(j = n-1, j > 0, j := j-1)$ **do**
  for $(i = 1, i \leq j, i := i+1)$ **do**
    $\{$ ha $A[i+1] < A[i]$, akkor cseréljük ki őket.$\}$

összehasonlítások száma: $n-1+n-2+\ldots+1 = \frac{n(n-1)}{2}$
cserék száma: $\frac{n(n-1)}{2}$

Kiválasztásos rendezés (Selection sort)

Input: $A[1:n]$ (rendezetlen) tömb
Keressük meg az $A[1:n]$ tömb minimális elemét és cserélük fel a tömb első elemével. Ezt rekurzívan ismételjük az $A[2:n], A[3:n], \ldots, A[n-1,n]$ tömbökre.

összehasonlítások száma: $n-1+n-2+\ldots+1 = \frac{n(n-1)}{2}$
cserék száma: $n-1$

Beszúrásos rendezés (Insertion sort)

Ha az $A[1:k]$ résztömb már rendezett, akkor szúrjuk be a következő elemet $A[k+1]$-et lineáris vagy bináris kereséssel, majd a következőt ebbe, stb.

	lineáris	bináris
összehasonlítás	$\frac{n(n-1)}{2}$	$\sum_{k=1}^{n-1} \lceil \log_2(k+1) \rceil$ $\approx n(\log_2 n - 0,442)$
mozgatás	$\frac{(n+2)(n-1)}{2}$	$\frac{(n+2)(n-1)}{2}$
átlagos összehasonlítás	$\frac{n(n-1)}{4}$	$\sum_{k=1}^{n-1} \lceil \log_2(n+1) \rceil$
átlagos mozgatás	$\frac{n^2}{4}$	$\frac{n^2}{4}$

Alsó becslés összehasonlítás alapú rendezésre

Ugyanaz, mintha Bar Kochba-ban kellene kitalálni, hogy az elemek melyik sorrendje (permutációja) az igazi sorrend.

Kezdetben $n!$ lehetséges sorrend jön szóba.
Két elemet összehasonlítva, a válasz két részre osztja a sorrendeket.
Ha pl. azt kapjuk, hogy $x < y$, akkor az olyan sorrendek, amikben x hátrébb van y-nál, már nem jönnek szóba.
Ha az ellenség megint úgy válaszol, hogy minél több sorrend maradjon meg, akkor k kérdés után még szóba jön $\frac{n!}{2^k}$ sorrend.
Ha $\frac{n!}{2^k} > 1$ nem tudjuk megadni a rendezést. $\Longrightarrow$

Tétel

Minden összehasonlítás alapú rendező módszer n elem rendezésekor legalább $\log_2(n!) \approx n \log_2 n$ *összehasonlítást használ.*

Összefésüléses rendezés (Merge sort)

Összefésülés (MERGE):

Két már rendezett sorozat (tömb, lista, stb.) tartalmának egy sorozatba való rendezése:
$A[1:k]$ és $B[1:l]$ rendezett tömbök $\Longrightarrow C[1:k+l]$ rendezett tömb
Nyilván $C[1] = \min\{A[1], B[1]\}$, pl. $A[1]$,
ezt rakjuk át C-be és töröljük A-ból.
$C[2] = \min\{A[2], B[1]\}$, stb.

Példa

A	B	C
12, 15, 20, 31	13, 16, 18	
15, 20, 31	13, 16, 18	12,
15, 20, 31	16, 18	12, 13
20, 31	16, 18	12, 13, 15
20, 31	18	12, 13, 15, 16
20, 31		12, 13, 15, 16, 18
31		12, 13, 15, 16, 18, 20
		12, 13, 15, 16, 18, 20, 31

összehasonlítások száma: $k + l - 1$, ahol k, l a két tömb hossza

Összefésüléses rendezés

Alapötlet: Rendezzük külön a tömb első felét, majd a második felét, végül fésüljük össze.
Ezt csináljuk rekurzívan.

$$\text{MSORT}(A[1:n]) :=$$
$$\text{MERGE}(\text{MSORT}(A[1:\lceil n/2 \rceil]), \text{MSORT}(A[\lceil n/2 \rceil + 1 : n])).$$

Hogy elvarrjuk a rekurzió alját, legyen $\text{MSORT}(A[i,i])$ az üres utasítás.

Összehasonlítások száma

Jelöljük $T(n)$-el a lépésszámot n hosszú tömb rendezésekor. Az egyszerűség kedvéért tegyük fel, hogy $n = 2^k$.

$$T(n) \le n - 1 + 2T(n/2),$$

$$T(n) \le n - 1 + 2(n/2 - 1 + 2T(n/4)) = n - 1 + 2(n/2 - 1) + 4T(n/4).$$

$$T(n) \le n-1+2(n/2-1)+4(n/4-1)+\cdots+2^{k-1}(n/2^{k-1}-1) \le n\lceil \log_2 n \rceil.$$

Felhasználva, hogy $T(1) = 0$.
Az összefésüléses rendezés konstans szorzó erejéig optimális.
Mozgatások száma: $2n\lceil \log_2 n \rceil$
Tárigény: $2n$ cella (bonyolultabban megcsinálva elég $n + \mathit{konst}$.)

Példa összefésüléses rendezésre

2	$\vert_3$	8	$\vert_2$	7	$\vert_4$	5	$\vert_1$	6	$\vert_6$	4	$\vert_5$	1	$\vert_7$	3
2		8	$\vert_2$	5		7	$\vert_1$	4		6	$\vert_5$	1		3
2		5		7		8	$\vert_1$	1		3		4		6
1		2		3		4		5		6		7		8

Gyorsrendezés (Quick sort)

[C. A. R. Hoare, 1960]
oszd meg és uralkodj: véletlen s elem a tömbből $\Longrightarrow$ PARTÍCIÓ(s) $\Longrightarrow$

s-nél kisebb elemek	s	$\dots$	s	*s-nél nagyobb elemek*

GYORSREND($A[1:n]$)
1. Válasszunk egy véletlen s elemet az A tömbből.
2. PARTÍCIÓ(s); az eredmény legyen az $A[1:k]$, $A[k+1:l]$, $A[l+1:n]$ felbontás.
3. GYORSREND($A[1:k]$); GYORSREND($A[l+1:n]$).

Véletlen elemnek választhatjuk mindig a tömb első helyén állót.

A PARTÍCIÓ(s) működése

Legyen $i := 1, j := n$,
$\Longrightarrow$ i-t növeljük, amíg $A[i] < s$ teljesül
$\Longrightarrow$ j-t csökkentjük, amíg $A[j] \geq s$
$\Longrightarrow$

	$i \rightarrow$	$\leftarrow j$
s-nél kisebb elemek		*s-nél nem kisebb elemek*

Ha mindkettő megáll (nem lehet továbblépni), és $i < j$, akkor $A[i] \geq s$ és $A[j] < s$ $\Longrightarrow$
Kicseréljük $A[i]$ és $A[j]$ tartalmát $\Longrightarrow$ $i := i + 1$ és $j := j - 1$. Ha a két mutató összeér (már nem teljesül $i < j$), akkor s előfordulásait a felső rész elejére mozgatjuk.
PARTÍCIÓ lépésszáma: cn
GYORSREND lépésszáma *legrosszabb esetben*: cn^2
GYORSREND lépésszáma *átlagos esetben*:
$1{,}39 n \log_2 n + cn = cn \log n$

Kulcsmanipulációs rendezések

Nem csak összehasonlításokat használ.
Pl. ismerjük az elemek számát, belső szerkezetét.

Ládarendezés (binsort)

Tudjuk, hogy $A[1:n]$ elemei egy m elemű U halmazból kerülnek ki, pl. $\in \{1, \dots, m\}$
$\Longrightarrow$ Lefoglalunk egy U elemeivel indexelt B tömböt (m db ládát), először mind üres.
első fázis $\Longrightarrow$ végigolvassuk az A-t, és az $s = A[i]$ elemet a $B[s]$ lista végére fűzzük.

$\Longrightarrow$ konzervatív rendezés, azaz az egyenlő elemek sorrendjét megtartja.

Ládarendezés (Bin sort)

Példa: Tegyük fel, hogy a rendezendő $A[1:7]$ tömb elemei 0 és 9 közötti egészek:

A:

5	3	1	5	6	9	6

B:

	1		3		5 5	6 6			9

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

B:

	1		3		5 5	6 6			9

A:

1	3	5	5	6	6	9

Lépésszám: B létrehozása $c_1 m$, első fázis $c_2 n$, második fázis $c_3(n+m)$, összesen $c_4(n+m)$.

Ez gyorsabb, mint az általános alsó korlát, ha pl. $m \leq cn$.