

# Adatszerkezetek és algoritmusok

HORVÁTH GÉZA

tizenharmadik előadás

# Előadások témái

- 1 Az algoritmusokkal kapcsolatos alapfogalmak bevezetése egyszerű példákon keresztül.
- 2 Az algoritmusok futási idejének aszimptotikus korlátai.
- 3 Az adatszerkezetekkel kapcsolatos alapfogalmak. A halmaz, a multihalmaz és a tömb adatszerkezet bemutatása.
- 4 Az adatszerkezetek folytonos és szétszórt reprezentációja. A verem, a sor és a lista.
- 5 Táblázatok, önátrendező táblázatok, hash függvények és hash táblák, ütközéskezelés.
- 6 Fák, bináris fák, bináris keresőfák, bejárás, keresés, beszúrás, törlés.
- 7 Kiegyensúlyozott bináris keresőfák: AVL fák.
- 8 Piros-fekete fák.
- 9 B-fák.
- 10 Gráfok, bejárás, legrövidebb út megkeresése.
- 11 Párhuzamos algoritmusok.
- 12 Eldönthetőség és bonyolultság, a P és az NP problémaosztályok.
- 13 **Lineáris idejű rendezés. Összefoglalás.**

# Leszámláló rendezés

A leszámoló rendezésnél feltételezzük, hogy az  $n$  bemeneti elem mindegyike 0 és  $k$  közötti egész szám.

COUNTING-SORT( $A, B, k$ )

```
1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 
```

# Leszámláló rendezés

	1	2	3	4	5	6	7	8
A	2	5	3	0	2	3	0	3

	0	1	2	3	4	5
C	2	0	2	3	0	1

(a)

	0	1	2	3	4	5
C	2	2	4	7	7	8

(b)

	1	2	3	4	5	6	7	8
B							3	

	0	1	2	3	4	5
C	2	2	4	6	7	8

(c)

	1	2	3	4	5	6	7	8
B		0					3	

	0	1	2	3	4	5
C	1	2	4	6	7	8

(d)

	1	2	3	4	5	6	7	8
B		0				3	3	

	0	1	2	3	4	5
C	1	2	4	5	7	8

(e)

	1	2	3	4	5	6	7	8
B	0	0	2	2	3	3	3	5

(f)

**Figure 8.2** The operation of COUNTING-SORT on an input array  $A[1..8]$ , where each element of  $A$  is a nonnegative integer no larger than  $k = 5$ . (a) The array  $A$  and the auxiliary array  $C$  after line 5. (b) The array  $C$  after line 8. (c)–(e) The output array  $B$  and the auxiliary array  $C$  after one, two, and three iterations of the loop in lines 10–12, respectively. Only the lightly shaded elements of array  $B$  have been filled in. (f) The final sorted output array  $B$ .

# Leszámláló rendezés

COUNTING-SORT( $A, B, k$ )

```

1  let  $C[0..k]$  be a new array
2  for  $i = 0$  to  $k$ 
3       $C[i] = 0$ 
4  for  $j = 1$  to  $A.length$ 
5       $C[A[j]] = C[A[j]] + 1$ 
6  //  $C[i]$  now contains the number of elements equal to  $i$ .
7  for  $i = 1$  to  $k$ 
8       $C[i] = C[i] + C[i - 1]$ 
9  //  $C[i]$  now contains the number of elements less than or equal to  $i$ .
10 for  $j = A.length$  downto 1
11      $B[C[A[j]]] = A[j]$ 
12      $C[A[j]] = C[A[j]] - 1$ 

```

A for ciklus a 2–3 sorban  $\mathcal{O}(k)$  lépést tesz meg. A for ciklus a 4–5 sorban  $\mathcal{O}(n)$  lépést tesz meg. A for ciklus a 7–8 sorban  $\mathcal{O}(k)$  lépést tesz meg, és a for ciklus a 10–12 sorban  $\mathcal{O}(n)$  lépést tesz meg. Így a teljes algoritmus futási ideje  $\mathcal{O}(k + n)$ . Stabil rendezés. (10. sor)

# Radix rendezés

329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

**Figure 8.3** The operation of radix sort on a list of seven 3-digit numbers. The leftmost column is the input. The remaining columns show the list after successive sorts on increasingly significant digit positions. Shading indicates the digit position sorted on to produce each list from the previous one.

**RADIX-SORT**( $A, d$ )

- 1 **for**  $i = 1$  **to**  $d$
- 2     use a stable sort to sort array  $A$  on digit  $i$

# Radix rendezés

## Lemma

*Legyen adott  $n$  darab  $d$  jegyből álló szám, ahol a számjegyek legfeljebb  $k$  különböző értéket vehetnek fel. Ekkor a radix rendezés  $\Theta(d(n + k))$  idő alatt rendezi ezeket a számokat.*

# Irodalomjegyzék

