Horváth Géza

tizenharmadik előadás

rezetés Leszámláló rendezés Radix rendezés Irodalomjegyzék

Előadások témái

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

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

```
COUNTING-SORT (A, B, k)
    let C[0...k] be a new array
   for i = 0 to k
        C[i] = 0
    for j = 1 to A. length
        C[A[j]] = C[A[j]] + 1
   // C[i] now contains the number of elements equal to i.
    for i = 1 to k
        C[i] = C[i] + C[i-1]
    /\!/ C[i] now contains the number of elements less than or equal to i.
10
    for j = A.length downto 1
        B[C[A[i]]] = A[i]
11
        C[A[i]] = C[A[i]] - 1
12
```

Leszámláló rendezés

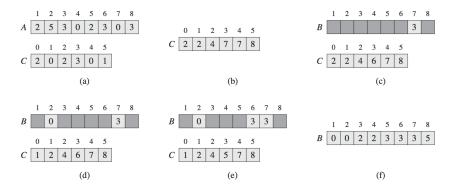


Figure 8.2 The operation of COUNTING-SORT on an input array A[1..8], where each element of A is a nonnegative integer no larger than k = 5. (a) The array A and the auxiliary array C after line S. (b) The array C after line S. (c)–(e) The output array B and the auxiliary array C after one, two, and three 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.

```
COUNTING-SORT (A, B, k)
    let C[0..k] be a new array
    for i = 0 to k
        C[i] = 0
   for i = 1 to A. length
        C[A[j]] = C[A[j]] + 1
    // C[i] now contains the number of elements equal to i.
    for i = 1 to k
        C[i] = C[i] + C[i-1]
    // C[i] now contains the number of elements less than or equal to i.
    for j = A.length downto 1
11
        B[C[A[j]]] = A[j]
        C[A[i]] = C[A[i]] - 1
12
```

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) Debrecen, 2023

Radix rendezés

329		720		720		329
457		355		329		355
657		436		436		436
839	jjjj	457	·····j]p·	839	jjp-	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

