

Линейные сортировки

Булгаков Илья, Гусев Илья

Московский физико-технический институт

Москва, 2020

Содержание

- 1 Виды сортировок
- 2 Сортировки за линейное время
- 3 Сортировка подсчетом (CountingSort)
- 4 Поразрядная сортировка или цифровая (RadixSort)

Виды сортировок

Напомним классификацию

- Сортировки сравнением
 - Особенность категории - в лучшем случае за $O(n \log(n))$
 - 1 Вставками (InsertionSort)
 - 2 Слиянием (MergeSort)
 - 3 Быстрая (QuickSort)
 - 4 С помощью кучи (HeapSort)
 - 5 ...
 - Сортировки без явного сравнения элементов
 - 1 Сортировка подсчетом (CountingSort)
 - 2 Сортировка поразрядная или цифровая (RadixSort)
 - LSD
 - MSD

Сортировка подсчетом (CountingSort)

Идея

Основная идея: для каждого входного элемента x определить количество элементов, которые меньше x .

С помощью этой информации элемент x можно разместить в той позиции выходного массива, где он должен находиться.

Например, если всего имеется 17 элементов, которые меньше x , то в выходной последовательности элемент x должен занимать 18 позицию.

Сортировка подсчетом (CountingSort)

Код

COUNTING-SORT(A, B, k)

```
1 Пусть  $C[0..k]$  — новый массив
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 // Сейчас  $C[i]$  содержит количество элементов, равных  $i$ .
7 for  $i = 1$  to  $k$ 
8    $C[i] = C[i] + C[i - 1]$ 
9 // Сейчас  $C[i]$  содержит количество элементов, не превышающих  $i$ .
10 for  $j = A.length$  downto 1
11    $B[C[A[j]]] = A[j]$ 
12    $C[A[j]] = C[A[j]] - 1$ 
```

Сортировка подсчетом (CountingSort)

Пример

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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 |

(б)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B | | | | | | | 3 | |

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

(в)

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

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

(г)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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 |

(д)

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

(е)

Поразрядная сортировка или цифровая (RadixSort)

Задача

Имеем множество последовательностей одинаковой длины, состоящих из элементов, на которых задано отношение линейного порядка. Требуется отсортировать эти последовательности в лексикографическом порядке.

Идея

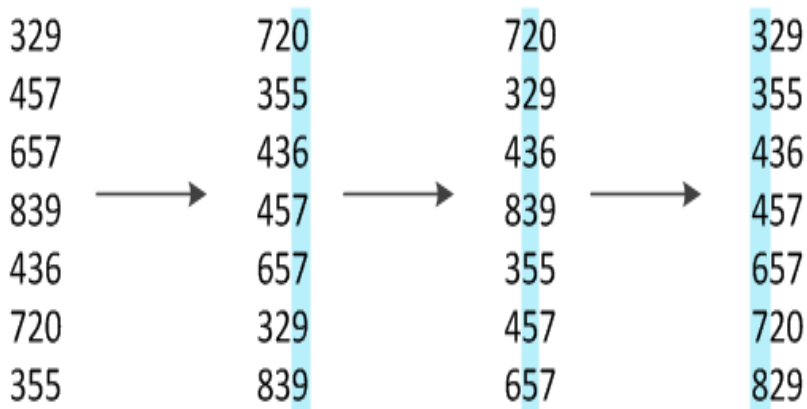
Сортируем поразрядно. На каждом шаге применяем устойчивую сортировку. Например, сортировку подсчетом.

Разновидности поразрядной сортировки

- LSD - Least Significant Digit radix sort
- MSD - Most Significant Digit radix sort

Поразрядная сортировка

Пример LSD



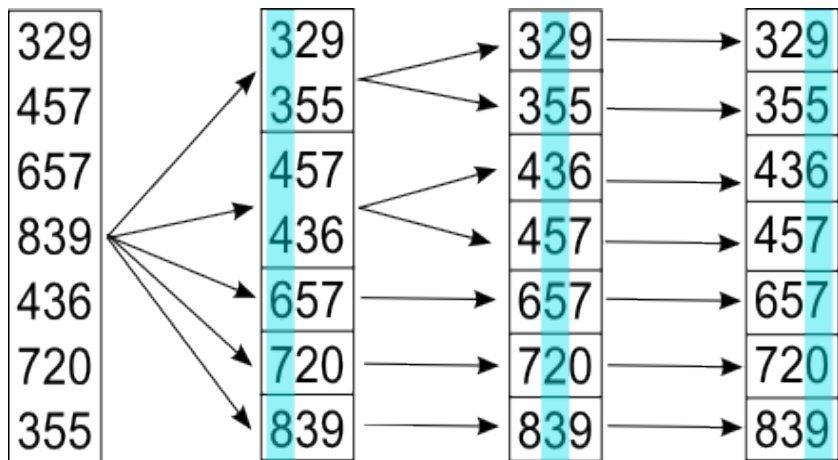
Поразрядная сортировка

Код LSD

```
function radixSort(int[] A):  
    for i = 1 to m  
        for j = 0 to k - 1  
            C[j] = 0  
        for j = 0 to n - 1  
            d = digit(A[j], i)  
            C[d]++  
        count = 0  
        for j = 0 to k - 1  
            tmp = C[j]  
            C[j] = count  
            count += tmp  
        for j = 0 to n - 1  
            d = digit(A[j], i)  
            B[C[d]] = A[j]  
            C[d]++  
    A = B
```

Поразрядная сортировка

Пример MSD




Поразрядная сортировка

Код MSD

```
function radixSort(int[] A, int l, int r, int d):
    if d > m or l >= r
        return
    for j = 0 to k + 1
        cnt[j] = 0
    for i = l to r
        j = digit(A[i], d)
        cnt[j + 1]++
    for j = 2 to k
        cnt[j] += cnt[j - 1]
    for i = l to r
        j = digit(A[i], d)
        c[l + cnt[j]] = A[i]
        cnt[j]--
    for i = l to r
        A[i] = c[i]
    radixSort(A, l, l + cnt[0] - 1, d + 1)
    for i = 1 to k
        radixSort(A, l + cnt[i - 1], l + cnt[i] - 1, d + 1)
```

Полезные ссылки I

 [Университет ИТМО: Цифровая сортировка](https://bit.ly/35wM87g)
<https://bit.ly/35wM87g>