SORTING in Linear Time:

Radix sort with

- distributing sort (for linked lists)
- counting sort (for arrays)

04.27.
2021

Radix Sort for Linked Lists

- The keys must be unsigned integers: r = number base, d = number of digits

In the examples: r = 4

$L = \langle 102, 313, 030, 010, 332, 103 \rangle$ (d = 3)

Buckets:
- $B_0 = \langle 030, 010 \rangle$
- $B_1 = \langle \rangle$
- $B_2 = \langle 102, 332 \rangle$
- $B_3 = \langle 313, 103 \rangle$

distributing sort according to digit 1 (rightmost digit, least significant digit)

$L = \langle 030, 010, 102, 332, 313, 103 \rangle$

- $B_0 = \langle 102, 103 \rangle$
- $B_1 = \langle 010, 313 \rangle$
- $B_2 = \langle \rangle$

distr. sort acc. to digit 2 (from the right)

$B_3 = \langle 030, 332 \rangle$

$L = \langle 102, 103, 010, 313, 030, 332 \rangle$ — SORTED acc. to rightmost 2 digits.

$B_0 = \langle 010, 030 \rangle$

$B_1 = \langle 101, 103 \rangle$

$B_2 = \langle \rangle$

$B_3 = \langle 313, 332 \rangle$

duster, sort acc. to digit 3.

$B[0..r-1]$

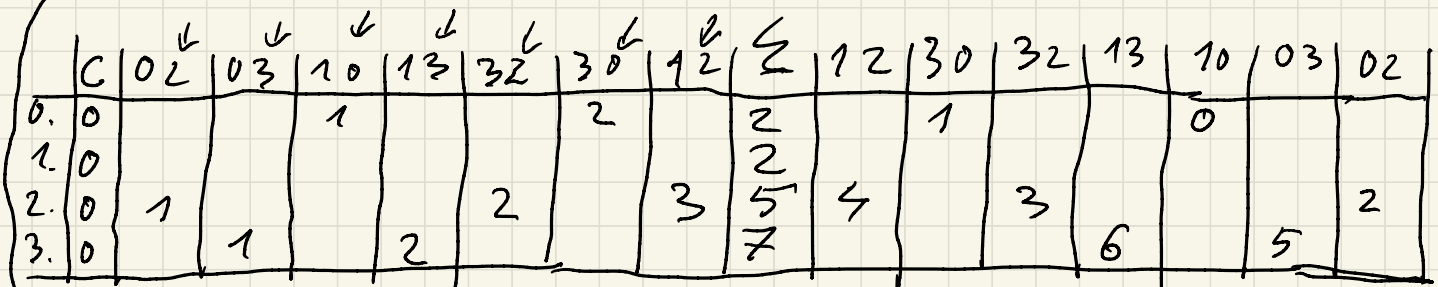
$L = \langle 010, 030, 101, 103, 313, 332 \rangle$ — SORTED (acc. to rightmost 3 digits ≈ all the digits)

$T_{\text{RadixSort}}(n, d, r) \in \Theta(d * (r + n)) = \Theta(n)$

In practice (d) and (r) are constants.

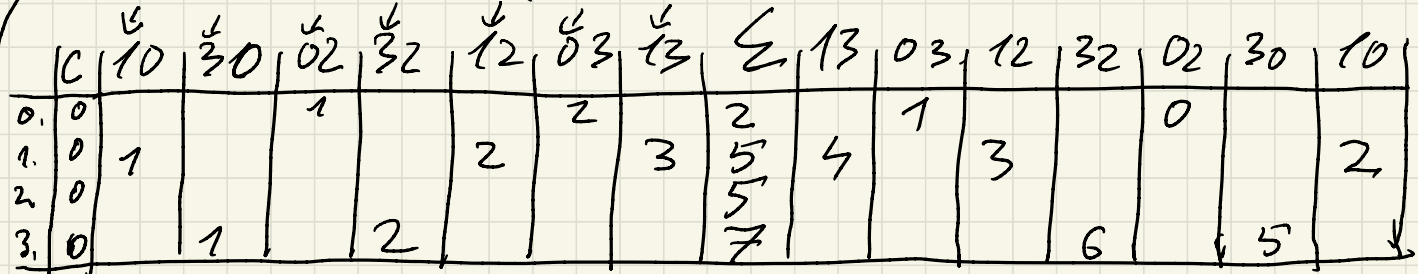
Radix Sort for Arrays

$A = \langle 02, 03, 10, 13, 32, 30, 12 \rangle$



Counting sort according to the 1. digits of the numbers

$B = \langle 10, 30, 02, 32, 12, 03, 13 \rangle$
0. 1. 2. 3. 4. 5. 6.



Counting sort acc. to the 2. digits of the numbers

$A = \langle 02, 03, 10, 12, 13, 30, 32 \rangle$ — SORTED ✓
0. 1. 2. 3. 4. 5. 6.

In programming often: $r = 256$ (1 byte is 1 digit)

One digit $\in 0..r-1$ $\quad d = 4 \vee d = 8$

$C[0..r-1]$ in the counter