

# Numeral systems & data structures

Jyrki Katajainen<sup>1)</sup>

Amr Elmasry<sup>2)</sup> and Claus Jensen<sup>3)</sup>

- 1) University of Copenhagen
- 2) Max-Planck-Institut für Informatik
- 3) The Royal Library

# Can you see the problem?

---

The decimal numeral system was introduced to the west through Muhammad ibn Mūsā al-Khwārizmī's book [al-Khwārizmī 825].

$$\begin{array}{r} 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \\ 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \quad 9 \\ + \\ \hline 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \end{array}$$

The binary numeral system has the same problem.

# Why is this a problem?

---

1 1 0 ... <sup>*i*</sup> 1 ... 1 0  
1 1 0 ... 1 ... 1 1



An addition of two bits can be a heavy operation!

# Goal

---

Let  $d$  be a positive integer.

$rep(d)$ :  $\langle d_0, d_1, \dots, d_{r-1} \rangle$  ( $d_0$  is the least significant digit)

$value(d)$ :  $\sum_{i=0}^{r-1} d_i \times w_i$  (in our case  $w_i = 2^i$ )

Develop a numeral system for which

- $\max \{d_i \mid i \in \{0, 1, \dots, r-1\}\}$  is as small as possible for all  $d$ ;
- an increment at any position  $i$  ( $increment(d, i)$ ) generates as few digit changes as possible in the worst case; and
- a decrement at any position  $i$  ( $decrement(d, i)$ ) generates as few digit changes as possible in the worst case.

# Strictly-regular system

---

**Digits:**  $d_i \in \{0, 1, 2\}$

**Strict regularity:** The sequence from the least-significant to the most-significant digit is of the form  $(1^+ | 01^*2)^* (\epsilon | 01^+)$ .

**Extreme digits:** 0 and 2.

Which of the following representations are strictly regular?

- a) 1111111
- b) 11011211101
- c) 1201
- d) 1110101

# Increment example

---

**Notation:** Digit  $d_i$  to be increased is displayed in **red**.  $d_a$  is the first extreme digit after  $d_i$ ,  $k$  is a positive integer,  $\alpha$  denotes any combination of  $1^+$  and  $01^*2$  blocks, and  $\omega$  any combination of  $1^+$  and  $01^*2$  blocks followed by at most one  $01^+$  block.

**Initial configuration:**  $\alpha 01^* \mathbf{1} 1^* 21^k \omega$

**Action:**  $d_i \leftarrow 2$ ;  $d_a \leftarrow d_a - 2$ ;  $d_{a+1} \leftarrow d_{a+1} + 1$

**Final configuration:**  $\alpha 01^* \mathbf{2} 1^* 021^{k-1} \omega$

# General algorithm

---

**Subroutine** *fix-carry*( $\mathbf{d}, i$ ): Assert that  $d_i \geq 2$ . Perform  $d_i \leftarrow d_i - 2$  and  $d_{i+1} \leftarrow d_{i+1} + 1$ .

---

**Algorithm** *increment*( $\mathbf{d}, i$ ):

---

- 1:  $++d_i$
  - 2: Let  $d_b$  be the first extreme digit before  $d_i$ ,  $d_b \in \{0, 2, \text{undefined}\}$
  - 3: Let  $d_a$  be the first extreme digit after  $d_i$ ,  $d_a \in \{0, 2, \text{undefined}\}$
  - 4: **if**  $d_i = 3$  **or** ( $d_i = 2$  **and**  $d_b \neq 0$ )
  - 5:     *fix-carry*( $\mathbf{d}, i$ )
  - 6: **else if**  $d_a = 2$
  - 7:     *fix-carry*( $\mathbf{d}, a$ )
-

# One of our results

---

**Theorem:** Given a strictly-regular representation of  $\mathbf{d}$ ,  $\text{increment}(\mathbf{d}, i)$  and  $\text{decrement}(\mathbf{d}, i)$  incur at most three digit changes.

**Proof:** By a case analysis. For  $\text{increment}(\mathbf{d}, i)$  we must consider 19 cases and for  $\text{decrement}(\mathbf{d}, i)$  15 cases. □

# Some related systems

---

**Regular system:** A regular sequence is of the form  $(0 \mid 1 \mid 01^*2)^*$ . Allows increments at any position with  $O(1)$  digit changes [Clancy & Knuth 1977].

**Extended regular system:**  $d_i \in \{0, 1, 2, 3\}$ . Every 3 is preceded by at least one  $\{0, 1\}$  before the next 3 or running out of digits, and every 0 is preceded by at least one  $\{2, 3\}$  before the next 0 or running out of digits. Allows increments and decrements at any position with  $O(1)$  digit changes [Clancy & Knuth 1977; Kaplan & Tarjan 1996].

# Full repertoire of operations

---

- increment*( $\mathbf{d}, i$ ): Assert that  $i \in \{0, 1, \dots, r\}$ . Perform  $++d_i$  resulting in  $\mathbf{d}'$ , i.e.  $value(\mathbf{d}') = value(\mathbf{d}) + w_i$ . Make  $\mathbf{d}'$  valid without changing its value.
- decrement*( $\mathbf{d}, i$ ): Assert that  $i \in \{0, 1, \dots, r - 1\}$ . Perform  $--d_i$  resulting in  $\mathbf{d}'$ , i.e.  $value(\mathbf{d}') = value(\mathbf{d}) - w_i$ . Make  $\mathbf{d}'$  valid without changing its value.
- cut*( $\mathbf{d}, i$ ): Cut  $rep(\mathbf{d})$  into two valid sequences having the same value as the numbers corresponding to  $\langle d_0, d_1, \dots, d_{i-1} \rangle$  and  $\langle d_i, d_{i+1}, \dots, d_{r-1} \rangle$ .
- concatenate*( $\mathbf{d}, \mathbf{d}'$ ): Concatenate  $rep(\mathbf{d})$  and  $rep(\mathbf{d}')$  into one valid sequence that has the same value as  $\langle d_0, d_1, \dots, d_{r-1}, d'_0, d'_1, \dots, d'_{r'-1} \rangle$ .
- add*( $\mathbf{d}, \mathbf{d}'$ ): Construct a valid sequence  $\mathbf{d}''$  such that  $value(\mathbf{d}'') = value(\mathbf{d}) + value(\mathbf{d}')$ .

# Conclusions

---

- We got the strictly-regular system from God; it was not invented by us.
- It is still open whether the system can be extended for ternary numbers ( $w_i = 3^i$ ).
- Also, it is open whether there exists an equally economical system that allows increments and decrements in  $O(1)$  worst-case time (we talked about the number of digit changes, not actual running time).

## Further reading

---

A. Elmasry, C. Jensen, and J. Katajainen, Strictly-regular number system and data structures, *Proceedings of 12th Scandinavian Symposium and Workshops on Algorithm Theory*, Lecture Notes in Computer Science **6139**, Springer-Verlag (2010)

A. Elmasry, C. Jensen, and J. Katajainen, The magic of a number system, *Proceedings of the 5th International Conference on Fun with Algorithms*, Lecture Notes in Computer Science, Springer-Verlag (2010)