

3b) Vom folosi algoritmi pentru lucrul cu numere mari

Funcție: Citire - Nr Mare()

\leftarrow citește NrMare (string)

pentru $i \leftarrow 0, \text{lungime}(\text{NrMare}) - 1$, execută

$\text{NrMare} \leftarrow \text{int}(\text{NrMare}[i])$

sfpentru

st = 0

dr = $\text{lungime}(\text{NrMare}) - 1$

cât timp st < dr execută:

aux = $\text{NrMare}[st]$

$\text{NrMare}[st] = \text{NrMare}[dr]$

$\text{NrMare}[dr] = \text{aux}$

st \leftarrow st + 1

dr \leftarrow dr - 1

sfcat timp

\leftarrow mută elementele din NrMare cu o poziție la dreapta

$\text{NrMare}[0] = \text{lungime}(\text{NrMare})$

return NrMare

sfFuncție

Funcție: aduna - NrMare()

dacă $a[0] > b[0]$ atunci

cât timp $a[0] \neq b[0]$ execută

\leftarrow adagă la finalul lui b pe 0

$b[0] \leftarrow b[0] + 1$

sfcat timp

sf dacă

altfel dacă $b[0] > a[0]$ atunci

cât timp $a[0] \neq b[0]$ execută
cât timp la primul lui a pe 0
 $a[0] \leftarrow a[0] + 1$
sf cât timp

$i \leftarrow 1$

$t \leftarrow 0$

$nr2 = []$

cât timp ($i \leq a[0]$) execută

$nr2[i] = (a[i] + b[i] + t) \% 10$
 $t = \lfloor (10a[i] + b[i] + t) / 10 \rfloor$
 $i \leftarrow i + 1$
sf cât timp

dacă $t \neq 0$ atunci

returnează 0 distanță cu -1 de maxime a[0]
sf dacă

return nr2

sf funcție

Dimensiunea problemei: $\{ \text{lungime}(a), \text{lungime}(b) \}$

Complexitate: (analog ca la comparare)

$$O(n) + O(\log_2(\text{lungime}(a, b))) + O(n) = O(n)$$