

3a) Vom folosi algoritmul pentru lucrul cu numere mari.

- Funcție: Citire — Nr Mare()

 <citește NrMare>(string)

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

 NrMare[i] = int(NrMare[i])

 sf pentru

 st=0

 dr=lungime(NrMare)-1

 — cât timp st < dr execută

 aux = NrMare[st]

 NrMare[st] = NrMare[dr]

 NrMare[dr] = aux

 st ← st + 1, dr ← dr - 1

 sf cât timp

 <mută elemente din NrMare cu o poziție la dreapta>

 NrMare[0] = lungime(NrMare)

return NrMare

sf Funcție

- Funcție: Compara NrMare(a,b)

 — dacă a[0] > b[0] atunci

 return "+1"

 sf dacă

 — dacă b[0] > a[0] atunci

 return "-1"

 sf dacă

 — pentru $i \leftarrow \text{lungime}(a)-1, 1, -1$ execută

 — dacă a[i] > b[i] atunci

 return "+1"

 — dacă b[i] > a[i] atunci

 return "-1"

 sf pentru

return 0 // Numere egale

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

Operațio dominantă: Avem 3 cicluri separate, deci 3 operații dominante

(i) - prima operație dominantă

(ii) - a doua operație dominantă

(iii) - a treia operație dominantă

(i) $\text{NrNare}[i] = \text{int}(\text{NrNare}[i])$

Estimare timp de execuție: $\sum_{i=1}^m 1 = m$
 $\Rightarrow O(m)$

(ii) $\text{aux} = \text{NrNare}[st]$

Estimare timp de execuție: $\left\lceil \log_2 \text{lungime}(\text{NrNare}) \right\rceil + 1$
 $\Rightarrow O(\log_2 \text{lungime}(\text{NrNare}) + 1)$

(iii) dacă $a[i] > b[i]$

Estimare timp de execuție: $\sum_{i=1}^{(m-1)} 1 = \frac{(1+1)(m-1)}{2} = m-1$
 $m = \text{lungime}(\text{NrNare})$

$0 \leq \text{dacă } a[i] > b[i] \leq m-1$

$\Rightarrow O(m)$

Complexitate: $(i) + (ii) + (iii) = O(m) + O(\log_2(\text{lungime}(\text{NrNare}) + 1)) + O(m)$
 $= O(m)$