

3d) Vom folosi algoritmul pentru înmulțirea a două numere mari

< Funcție: Căline - NrMare()

< analog ca la subpunctele a, b, c >

— Funcție: înmulțire - NrMare(a, b)

prod = []

prod[0] = (a[0] + b[0] - 1)

— pentru $i \leftarrow 1, b[i] + 1$ execută:

— pentru $j \leftarrow 1, a[j] + 1$ execută

prod[i+j-1] = (b[i] * a[j])

sf pentru

sf pentru

t ← 0

— pentru $i \leftarrow 1, \text{prod}[i] + 1, 1$ execută

t ← t + prod[i]

prod[i] = t % 10

t = [t / 10]

sf pentru

— dacă t ≠ 0 atunci

// prod[prod[0]+1] = t

returnează < o listă cu -1 de lungime a[0] + b[0] >

sf dacă

returnează prod

sf Funcție

Dimensiunea problemei: $\text{card}(C_{k_1} C_{k_1-1} C_{k_1-2} \dots C_0, C_{k_2} C_{k_2-1} C_{k_2-2} \dots C_0)$

Operatie dominantă: Sunt mai multe, dar vai lua în calcul doar cea care are cea mai mare complexitate.

$$\text{Estimare timp de execuție: } \sum_{i=1}^m \sum_{j=1}^n 1 = \sum_{i=1}^n n = \frac{2n \cdot n}{2} = \frac{2n^2}{2}$$

$$\Rightarrow O(n^2) \text{ caz particular } m=n$$

$$\Rightarrow O(mn) \text{ caz general}$$

$$\text{Complexitate: } \begin{cases} O(n^2) - \text{particular } m=n \\ O(mn) - \text{general} \end{cases}$$