

A. subalgoritm suma(n)

$\times 0$

pentru $k = 1, n \times n \times n$ execuție

$\times 0 + 1$

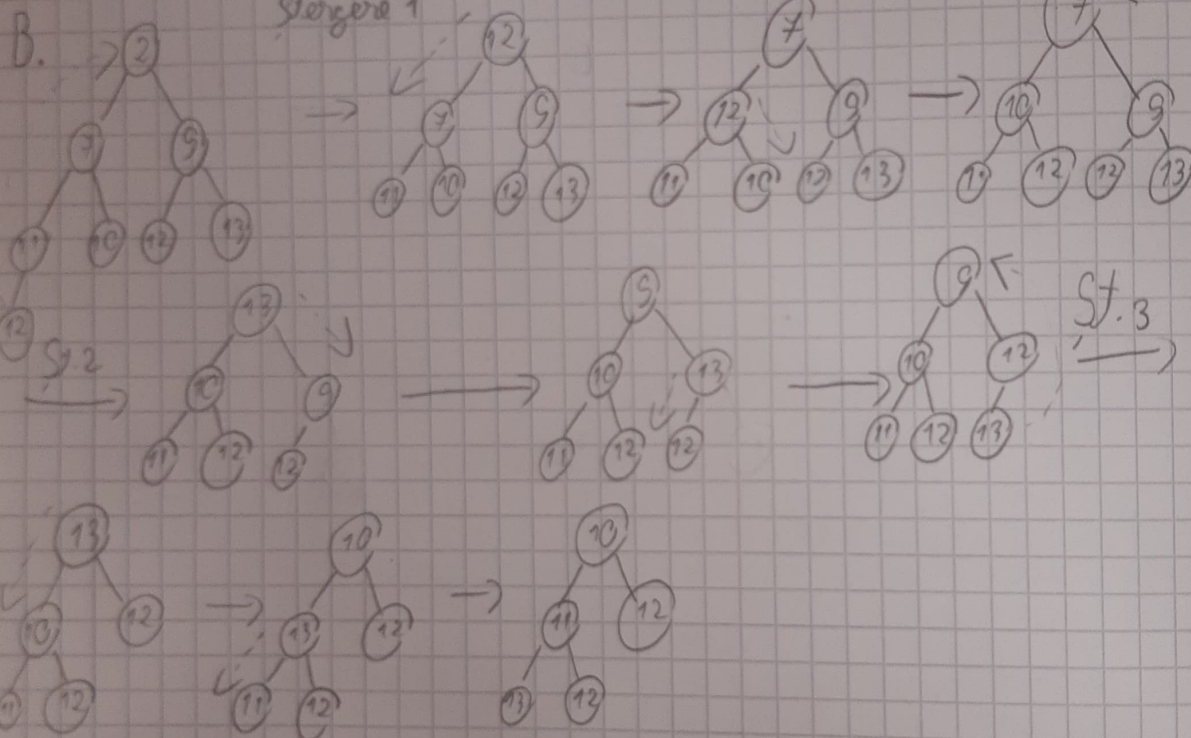
stăruiește pentru

① timparele

stăruiește subalgoritm

$$T(n) = 1 + n^3 \in \Theta(n^3) \in O(n^3)$$

stăruiește 1



C. $O(n)$

În buchet sort avem complexitatea timp $\Theta(n+d)$, unde d e cel mai mare el. (2000)

$$\Theta(n+d) = \Theta(n+2000) \in \Theta(n)$$

C2. ① Complexitatea în cel mai rău caz este $\Theta(n)$ (sau $\Theta(\log_2 n)$ pentru AVL)

Container:

rad: \uparrow Rad

Nod: st: \uparrow Nod

dr: \uparrow Nod

e: \uparrow Element

subalgoritm get cod(a^i, e', cod^o)

cr \leftarrow a.rad

cod \leftarrow "1"

ok \leftarrow F

cât timp cr \neq NIL execută

dacă [cr].e = e atunci

ok \leftarrow T

@STOP

altfel

dacă [cr].e \leq e atunci

cr \leftarrow [cr].st; cod \leftarrow cod + "0"

altfel

cr \leftarrow [cr].dr; cod \leftarrow cod + "1"

sfasit dacă

sfasit dacă

sfasit cât timp

Dacă ok = T atunci

@ Nodul nu a fost găsit

sfasit dacă

sfasit subalgoritm