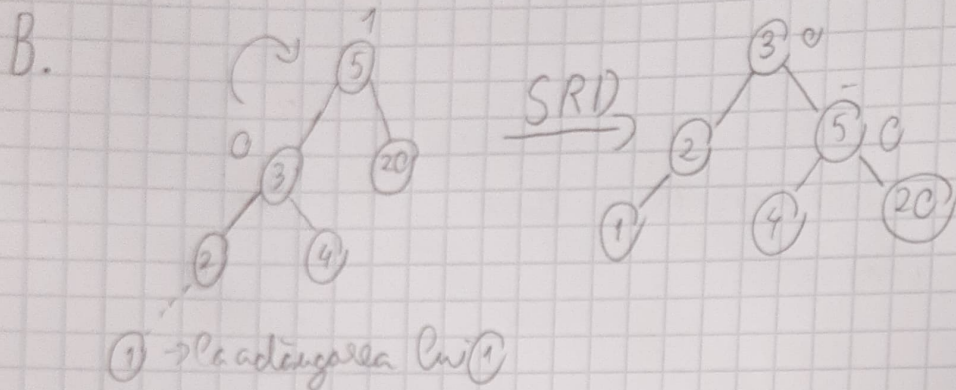


$$A. T(n) = \begin{cases} 1, & n=1 \\ 1+T(n/2), & \text{altfel} \end{cases}$$

$n=2^k (\dots)$

Complexitate $\Theta(\log_2 n)$ in ambele cazuri

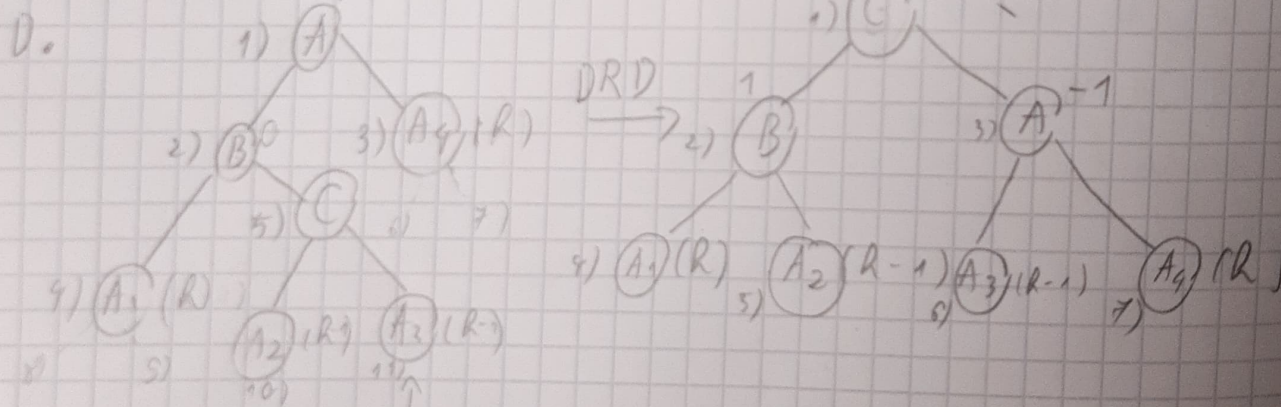


C. a) b) d) -> algoritmi care sort funcționare pe baza de nr.

-> Radix Sort: considerăm n nr. maxime de zecimale de după virgulă

-> 11, 7, 5, 6 ($\Rightarrow \{11, 7, 5, 6\}$) -> fiecare nr. îl transformăm în astfel de tuple

C2. a) data [1]



A_1 nu își va schimba poziția în ansamblu

A_2 : dacă un descendent e pe poziția $i \Rightarrow$ noua poziție va fi $i-6$

A_3 : dacă un descendent e pe poziția $i \Rightarrow$ noua poziție va fi $i-5$

A_4 : dacă un descendent e pe poziția $i \Rightarrow$ noua poziție va fi $i+9$

subalgoritm schimbă (v, pos, dif, qa)

{ algoritm ajutător care adaugă fiecarei excedențe din ansamblu dif }

crearea (c) { coadă }

cât timp $v \neq NIL$ execută

sterge(v, i);

$v[i+1] \leftarrow a \cdot v[i]$ $i = i + 1$

Dacă $2 \cdot i \leq a \cdot n \wedge a \cdot v[2 \cdot i] \neq NIL$ atunci

adăugă($v, 2 \cdot i$)

sfârșit dacă

Dacă $2 \cdot i + 1 \leq a \cdot n \wedge a \cdot v[2 \cdot i + 1] \neq NIL$ atunci

adăugă($v, 2 \cdot i + 1$)

sfârșit dacă

sfârșit cât timp

sfârșit subalgoritm

Container:

$v: T[Elem[]]$

$n: \text{Întreg}$

$max: \text{Întreg}$

Subalgoritm $DRD(a)$

PRE: a : exlore binde către

POST: $a \cdot v$ finizat dublu la dreapta

pentru $i = 1, n$ execută

$v[i] = NIL$

sfârșit pentru

$v[1] = a \cdot v[5] \{C\}$

$v[2] = a \cdot v[2] \{B\}$

$v[3] = a \cdot v[1] \{A\}$

$schimbă(v, 8, 0, a) \{A_1\}$

$schimbă(v, 10, -6, a) \{A_2\}$

$schimbă(v, 11, -5, a) \{A_3\}$

$schimbă(v, 3, 7, a) \{A_4\}$

pentru $i = \overline{1, n}$ execută

$$a. v[1] \leftarrow v[i]$$

A.

sfârșit pentru

Sfârșit subalgoritm

$\rightarrow C$

B.

C.

\rightarrow

\rightarrow

C₂

D.