

A.  $\text{Caz fav} = \text{Caz mediu} = \text{Caz defav} \Rightarrow \text{Timpi mediu și defav. sunt egali}$

$$n = 2^k \Rightarrow k = \log_2 n \quad ; \quad T(n) = \begin{cases} 1, & n = 1 \\ T(n/2) + 1, & n > 1 \end{cases}$$

$$T(2^k) = 1 + T(2^{k-1})$$

$$T(2^{k-1}) = 1 + T(2^{k-2})$$

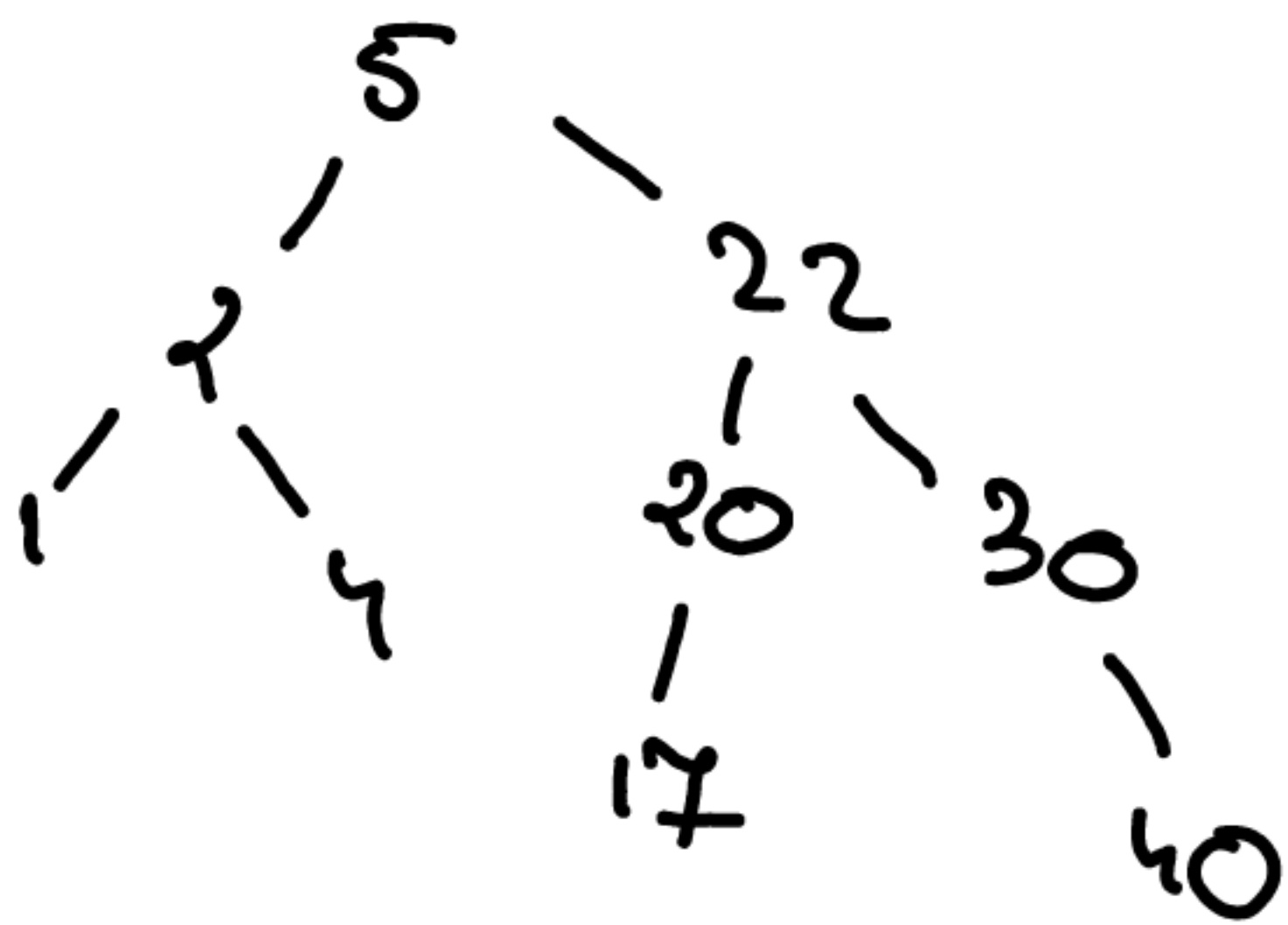
$$\vdots$$
$$T(1) = 1$$

$$T(2^k) = 1 + \dots + 1$$

$$T(2^k) = k + 1$$

$$T(n) = \log_2 n + 1 \in \Theta(\log_2 n)$$

B.



- se va alege maximum din subarborul stâng

C1. a)

Lista implementată secvențial este cea mai potrivită, deoarece elementele de pe o poziție dată pot fi accesate în timp constant  $\Theta(1)$ , spre deosebire de celelalte opțiuni care au nev. de timp liniar.

c2. c)

- deoarece înălțimea unui nod e reprez. de lungimea drumului cel mai lung de la un nod la o frunză, din subarborii în care nodul e rădăcină (drumul: 8, 10, 11)

D.

Arbore

$e: TElement[]$

$st: Integ23$

$dr: Integ23$

$n: Integ$  (poziția pe care se află rădăcina)

Subprogram BFS ( $a, e, e'$ ) este

{ par:  $a \in Arbore$

$e \in TElement$

$e' \in TElement$

post: se returnează - 1 dacă  $e$  și  $e'$  sunt pe  
același nivel  
- 0 în caz contrar

{  
nivel1  $\leftarrow -1$ ; nivel2  $\leftarrow -1$

Dacă  $n \neq -1$  atunci

crează( $c$ ) { se creează coada }

aduga( $c, \{x, 0\}$ )

Cât timp  $n \neq 0$  execută

{ se extrag indexul și nivelul nodului }

store( $c, \{i, nivel\}$ )

Dacă  $a.e[i] = e$  atunci

nivel1  $\leftarrow nivel$

și dacă



Dacă  $a.ezi \neq e$  atunci  
nivel  $2 \leftarrow nivel$

SR Dacă

se adaugă descendenți?

Dacă  $a.stzi \neq -1$  atunci

adaugă  $(c, \{a.stzi, nivel + 1\})$

SR Dacă

Dacă  $a.dz \neq -1$  atunci

adaugă  $(e, \{a.dz, nivel + 1\})$

SR Dacă

SR cât timp

SR Dacă

Dacă  $nivel_1 = nivel_2 \wedge nivel_1 \neq -1$  atunci

BFS  $\leftarrow$  "da"

afire

BFS  $\leftarrow$  "nu"

SR Dacă

SR Subalgoritm

Complexitate — timp:  $\Theta(n)$

— spațiu:  $\Theta(n)$   
(aditional)

considerând că op. de

adaugare și ștergere efectuate de coadă se execută în  
timp constant  $\Theta(1)$