A. Subalgoritm $lg(n)$:

    dacă $n = 1$ atunci

        $lg \leftarrow 1$

    altfel

        $lg \leftarrow lg(n/2)$

    sfârșit dacă

sfârșit subalgoritm

B. 3 (Lungimea lanțului maxim până la o frunză)

C. b)

C2. d)

D.

Container:

    n: Întreg, nr de elemente   indexat de la 1 $\prec$: ↑Element × ↑Element → {A, F}

                                relația de ordine

    v: ↑Elem[], vectorul cu legături

    max: Întreg, capacitatea maximă de memorare

subalgoritm (á, č)

    a: un ansamblu
        ↑Element

    e: elementul cel mai prioritar, șters (rădăcina)

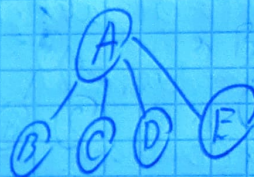   @ aruncă excepție dacă ansamblul e vid

Dacă $a.n = 0$ atunci

    @ ansamblu vid

sfârșit dacă

    $e \leftarrow a.v[1]$; $aux \leftarrow a.v[a.n]$

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

    $a.n \leftarrow a.n - 1$

=> $A \prec B$
   $A \prec C$
   $A \prec D$
   $A \prec E$

{ coborâm rădăcina }

current ← 1 ; OK ← Adevărat

cât timp 4×current ≤ a.n și OK execută

 i ← 4 × current ; j ← i

 Dacă i+1 ≤ a.n și a.r(a.v[i+1], a.v[j]) atunci

  j ← i+1

 sfârșit dacă

 Dacă i+2 ≤ a.n și a.r(a.v[i+2], a.v[j]) atunci

  j ← i+2

 sfârșit dacă

 Dacă i+3 ≤ a.n și a.r(a.v[i+3], a.v[j]) atunci

  j ← i+3

 sfârșit dacă

 Dacă nu a.r(aux, a.v[j]) execută

  OK ← Adevărat

  a.v[current] ← a.v[j]

  current ← j

 altfel

  OK ← Fals

 sfârșit dacă

sfârșit cât timp

a.v[current] ← aux

sfârșit subalgoritm