

A. Scrieți un subprogram recursiv având complexitatea timp $O(n^2)$. Justificați complexitatea (prin deducția acesteia).

Subprogram $f(n, i)$

Dacă $n=0$ atunci

$f \leftarrow 0$

Altfel

Return $i=1, n$ execută
~~Tipărește~~ 0

Altfel

$f \leftarrow f(n-1)$

Altfel

$$T(n) = \begin{cases} 1, & n=0 \\ n+T(n-1), & n>0 \end{cases}$$

$$T(n) = n + T(n-1)$$

$$T(n-1) = n-1 + T(n-2)$$

$$T(n-2) = n-2 + T(n-3)$$

$$T(n-3) = n-3 + T(n-4)$$

$$T(n-4) = n-4 + T(n-5)$$

$$T(n-5) = n-5 + T(n-6)$$

$$T(n-6) = n-6 + T(n-7)$$

$$T(n-7) = n-7 + T(n-8)$$

$$T(n-8) = n-8 + T(n-9)$$

$$T(n-9) = n-9 + T(n-10)$$

$$T(n-10) = n-10 + T(n-11)$$

$$T(n-11) = n-11 + T(n-12)$$

$$T(n-12) = n-12 + T(n-13)$$

$$T(n-13) = n-13 + T(n-14)$$

$$T(n-14) = n-14 + T(n-15)$$

$$T(n-15) = n-15 + T(n-16)$$

$$T(n-16) = n-16 + T(n-17)$$

$$T(n-17) = n-17 + T(n-18)$$

$$T(n-18) = n-18 + T(n-19)$$

$$T(n-19) = n-19 + T(n-20)$$

$$T(n-20) = n-20 + T(n-21)$$

$$T(n-21) = n-21 + T(n-22)$$

$$T(n-22) = n-22 + T(n-23)$$

$$T(n-23) = n-23 + T(n-24)$$

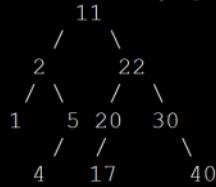
$$T(n-24) = n-24 + T(n-25)$$

$$T(n) = n + n-1 + \dots + 1 + 1$$

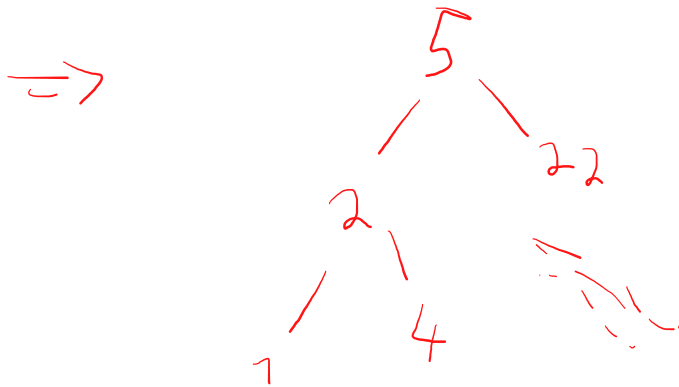
$$= 1 + 2 + \dots + n + 1$$

$$= \frac{n(n+1)}{2} + 1 \in O(n^2)$$

B. Fie următorul ABC. Presupunând ca vrem să ștergem rădăcina și să o înlocuim cu ceva din subarborele stâng, care va fi arborele rezultat în urma ștergerii? Justificați



La ștergerea unui nod cu 2 fi se ia nodul maxim din subarborele stâng $\Rightarrow 5$



C. Într-un ansamblu construit cu relația \leq , operația de accesare a unui element se referă la:

- a) orice element din ansamblu b) elementul cu valoare minimă c) elementul cu valoare maximă
Justificați

În informatică, dacă avem un **ansamblu de date** (ex: heap, mulțime ordonată, coadă cu priorități etc.) construit cu o **relație de ordine** – de exemplu, relația $<$, atunci:

- Structura va păstra elementele **ordonate crescător** (implicit)
- „Accesarea” unui element înseamnă, în mod normal, **accesarea elementului „de top”** în funcție de acea relație

✓ Ce înseamnă „operația de accesare”?

Este echivalentul cu:

- „Peek” într-un heap
- „Front” într-o coadă cu priorități
- „Minim” într-o mulțime ordonată

C. Considerăm expresia în forma infixată: $(6*3-12)*2-(3+7)$. Presupunem că folosim o Stivă pentru a converti expresia din forma infixată în forma postfixată. Care este numărul maxim de simboluri care vor apărea în stivă la un moment dat de-a lungul conversiei? Justificați a) 1 b) 2 **c) 3** d) 4

conversiei? Justificati

a) 1

b) 2

c) 3

d) 4

STIVĂ: $(* \rightarrow (- \rightarrow \rightarrow * \rightarrow - \rightarrow -$ ~~))~~

FPP: $63 * 12 - 2 * 37 + -$