

A. Scrieți un subalgoritm iterativ care să aibă timpul de execuție  $\Theta(n^2 \log n)$ . Deduceți complexitatea. Nu se va folosi funcția matematică **logarithm**.

Subalgoritm  $f(n)$

Pentru  $i=1, n$  execută

Pentru  $j=1, n$  execută

$k \leftarrow 1$

În timp  $k \leq n$  execută

$k \leftarrow k * 2$

În timp

În timp

În timp

În timp

$$\sum_{k=0}^{\lfloor \log_2 n \rfloor} 2^k$$

$$\sum_{k=1}^{\log_2 n} 1$$

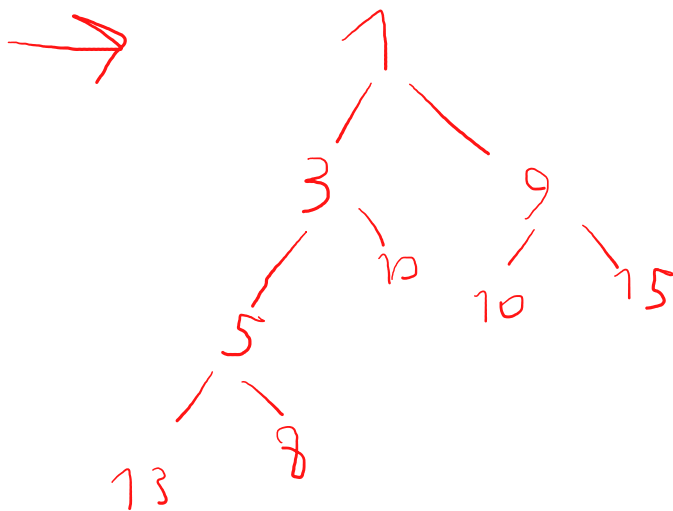
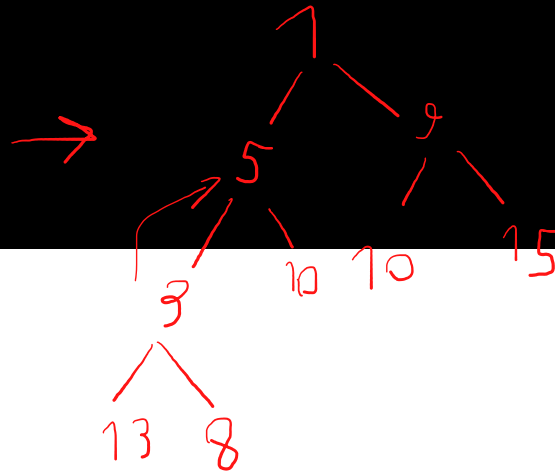
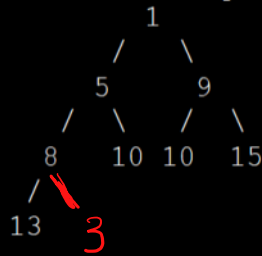
$$T(n) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{\log_2 n} 1$$

$$= \sum_{i=1}^n \sum_{j=1}^n \log_2 n$$

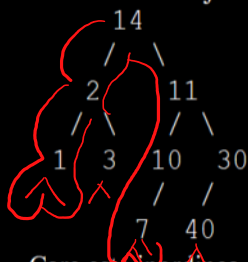
$$= \sum_{i=1}^n n \log_2 n$$

$$= n^2 \log_2 n \in \Theta(n^2 \log n)$$

B. Arătați ansamblul rezultat prin inserarea valorii 3 în următorul ansamblu. Justificați



C. Fie arborele de mai jos.



1 2 3 14 7 10 11 40 30

Care este înordinea arborelui? Justificați

a) 1 2 3 7 10 11 14 30 40

**b) 1 2 3 14 7 10 11 40 30**

c) 1 3 2 7 10 40 30 11 14

d) 14 2 1 3 11 10 7 30 40

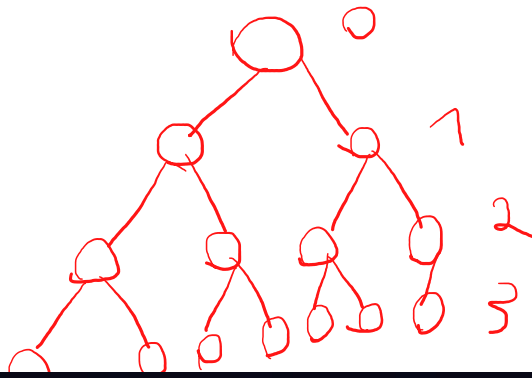
C. Presupunem că T este un arbore binar cu 14 noduri. Care este adâncimea minimă a lui T? Justificați

a) 0

**b) 3**

c) 4

d) 5



Adâncimea minimă a lui  $T$  este atunci când  $T$  este aproape plin

$$\rightarrow 2^0 + 2^1 + \dots + 2^h = 14$$

$$\Rightarrow 2^{h+1} - 1 = 14 \Rightarrow 2^{h+1} = 15 \Rightarrow h+1 = \log_2 15 \Rightarrow h+1 = 4 \text{ or } h = 3$$