

Busiac Andrei  
Gr. 212  
10

24-06-2021

Subiect 56

$$A. \quad T(n) = \begin{cases} 4 T\left(\frac{n}{2}\right), & n > 1 \\ 1, & \text{altfel} \end{cases}$$

$$T(n) = 4 T\left(\frac{n}{2}\right)$$

$$\begin{aligned} T(2^k) &= 4 T(2^{k-1}) = \\ &= 4^2 T(2^{k-2}) = \\ &= 4^3 T(2^{k-3}) = \\ &= \dots = 4^k T(1) = 4^k = \\ &= 2^{2k} = (2^k)^2 \end{aligned}$$

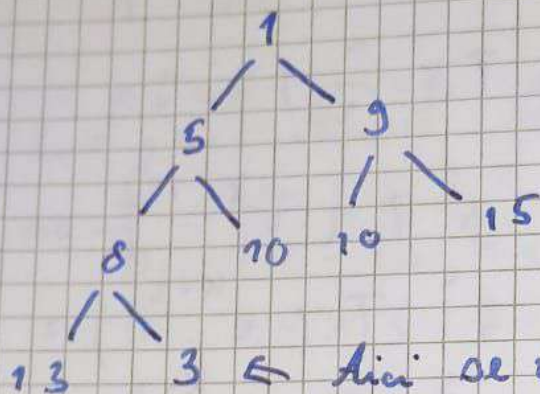
$$\Rightarrow T(n) = n^2$$

Dim. nouă că cazul defavorabil este  
egal cu cazul mediu, complexitatea  
va fi  $\Theta(n^2)$ .

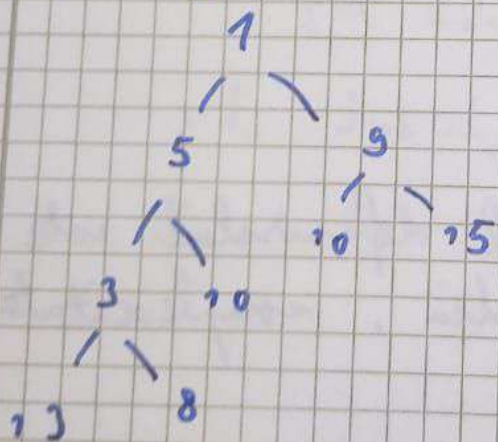


Crucian Andrei  
Ggr. 212  
10

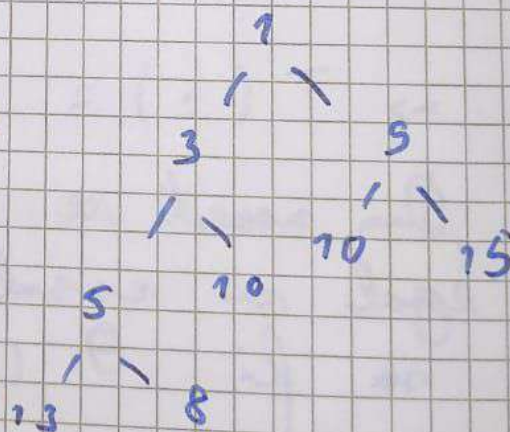
B. Initial, 3 se adaugă astfel încât structura de ansamblu să fie respectată (de la stînga la dreapta, pe ultimul nivel).



În toate acestea, nu este respectată proprietatea ansamblului (în acest caz, min-heap). Astfel, îl vom tot "sturna" pe 3.



→



Ansamblul rezultat



Curse Andrei

Gr. 212

K

C1. a) adâncimea arb.

c) înălțimea arb.

d = Înălțimea unui arbore, egală cu adâncimea sa, este nivelul maxim al nodurilor din arbore, iar în cazul ABC degenerati, d coincide cu numărul de noduri.

C2. d) 14      2      11      1      3      10      30

7      40

La parcurgerea pe niveluri, se iau nodurile de la stânga la dreapta pentru fiecare nivel, începând cu nivelul 0 (rădăcina).



Enrica Andrei

Cor. 2.12

Ko

D. Se potrivește arborii în ordine,  
ier într-o nodulă se încheiește un  
mărul asociat nodului curent. Când se gă-  
sește valoarea căută, menționăm nodulă.

Subalgoritmul în ordine (  $ab$ ,  $e$ ,  $n$  )

pre:  $ab$  arbore binar pe tablă,

$e$  TELEMENT

~~$n$  întreg~~

post:  $n$  întreg, numărul asociat nodului  
 $e$ , -1 altfel

{ vom folosi o stivă auxiliară, în  
care se vor menține ~~inițial~~ mai întotdeauna  
elementele de pe ramura stângă }

creșterea ( S )

{ inițializarea  $n$  cu -1 }

$n \leftarrow -1$

{ schimbăm arborii printre un vector,  
unde rădăcina e pe poz. 1 }

$r \leftarrow 1$



Busico Andrei  
Ex. 212  
de

} utilizăm o variabilă pentru incrementare  
 $K \leftarrow 0$

Cât timp  $\neg$  vidă (S)  $\vee$  (arb[ $i$ ]  $\neq$  ~~Hil~~ <sup>Hil</sup>)  
execută

  cât timp (arb[ $i$ ]  $\neq$  Hil) execută

    adaugă (S, arb[ $i$ ])

$i \leftarrow 2 \cdot i$

  } se adaugă fiecare stângă până la  
  epuizare }

sf cât timp

  sterge (S,  $\gamma$ )

$K \leftarrow K + 1$

  dacă  $\gamma = e$  atunci

$n \leftarrow K$

  sf Dacă

$i \leftarrow 2 \cdot i + 1$

  } se continuă cu partea dreaptă }

sf Cât timp

Complexitate:  $\Theta(n)$

sf. Subalgoritm.

p. 5