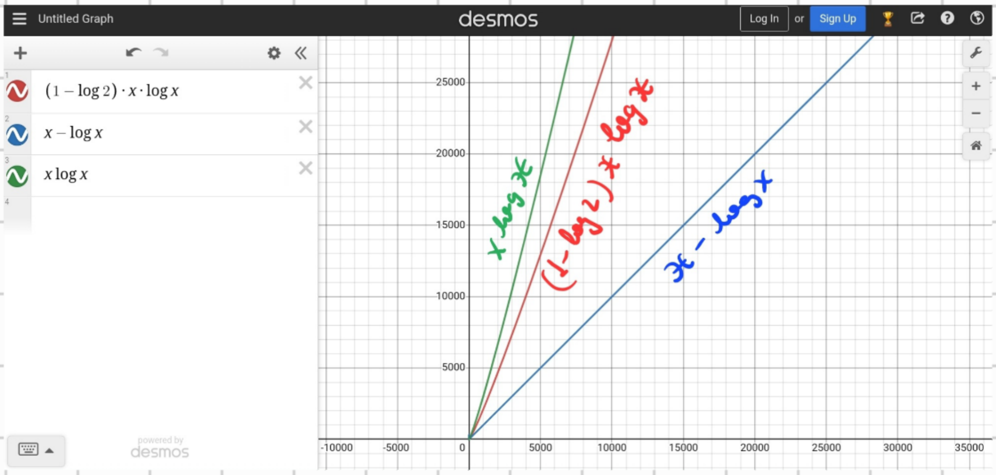
Temă seminar #4

1) complexitate pt Interclasare a 2 BST – varianta recursivă

$$T(n) = 2T\left(\frac{n}{2}\right) + O(\log n)$$

↳ nu se poate aplica T. Master ⇒ det. recursiv și calc. limitele

$$T(n) = 2T\left(\frac{n}{2}\right) + O(\log n) = ?$$



$$T(n) \sim n \log n$$

sau

$$T(n) = n \log n \,(1 - \log 2) + (n \cdot 2 \log 2 - \log n) - 2 \log 2$$

(o încercare de :)) DEMONSTRAȚIE :

$$t(n) = 2t\left(\frac{n}{2}\right) + \log n$$

$$t(n) = 2t\left(\frac{n}{2}\right) + O(\log n)$$

$$t(n) = 2t\left(\frac{n}{2}\right) + \log n$$

$$n \leftarrow \frac{n}{2} \Rightarrow t\left(\frac{n}{2}\right) = 2t\left(\frac{n}{4}\right) + \log\frac{n}{2} \Rightarrow$$

$$\Rightarrow t(n) = 2\left(2t\left(\frac{n}{4}\right) + \log\frac{n}{2}\right) + \log n$$

$$= 2\left(2\left(2t\left(\frac{n}{2^3}\right) + \log\frac{n}{2^2}\right) + \log\frac{n}{2}\right) + \log n$$

$$= 2^3 t\left(\frac{n}{2^3}\right) + 2^2\log\frac{n}{2^2} + 2\log\frac{n}{2} + \log n$$

$$= 2\left(2\left(2\left(2t\left(\frac{n}{2^4}\right) + \log\frac{n}{2^3}\right) + \log\frac{n}{2^2}\right) + \log\frac{n}{2}\right) + \log n$$

$$= 2^4 t\left(\frac{n}{2^4}\right) + 2^3\log\frac{n}{2^3} + 2^2\log\frac{n}{2^2} + 2\log\frac{n}{2} + \log n$$

$\vdots$

$$\log\frac{n}{2^{K-1}} \to \log 2 \Rightarrow n = 2^K$$
$$\Rightarrow K \sim \log n$$

$$= 2\, t(1) + \left(2^{K-1} + 2^{K-2} + \dots + 2^1 + 2^0\right)\log n -$$
$$- \sum_{i=0}^{K-1} 2^i \log 2^i =$$

$$\text{I)} \log n \sum_{i=0}^{K-1} 2^i \xrightarrow[\text{geom}]{\text{progr.}} \left(1 \cdot \frac{2^K - 1}{2-1}\right)\log n$$

$$\text{II)} -\log 2 \sum_{i=0}^{K-1} i\, 2^i =$$

$$1 + 2\cdot 2 + 3\cdot 2^3 + 4\cdot 2^4 + 5\cdot 2^5 + 6\cdot 2^6 + \dots$$

$$\sum_n n x^n = I \;/ :x$$

$$\sum_n n x^{n-1} = \frac{I}{x} \;/ \int (\forall) x$$

$$\sum_n \int \underbrace{n x^{n-1}}_{(x^n)'} dx = \int \frac{I}{x}$$

$$\sum_n x^n = \int \frac{I}{x}$$

$$\overset{x=2}{\Leftrightarrow} \sum_{n=0}^{K-1} x^n = \int \frac{I}{x}$$

$$\overset{x\neq 1}{\Leftrightarrow} \frac{x^K - 1}{x-1} = \int \frac{I}{x} \;/\; '$$

$$\frac{Kx^{K-1}(x-1) - x^K + 1}{(x-1)^2} = \frac{I}{x}$$

$$\frac{K\cdot x^K - K x^{K-1} - x^K + 1}{(x-1)^2} = \frac{I}{x}$$

$x \leftarrow 2$

$$K 2^K - K 2^{K-1} - 2^K + 1 = \frac{I}{2}$$

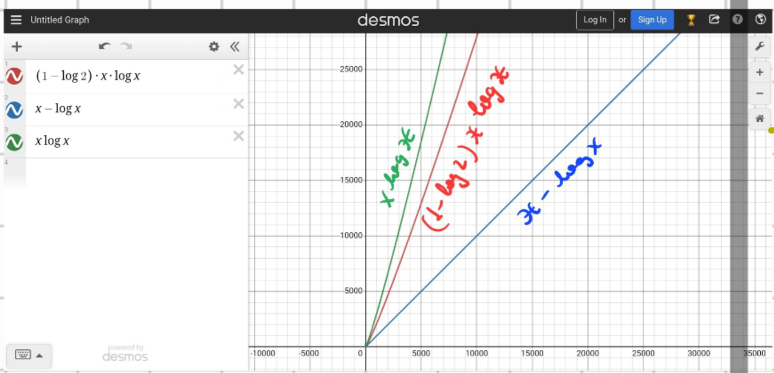
$$2[(K-1)2^K - K\cdot 2^{K-1} + 1] = I$$

$$I = (K-1)2^{K+1} - K 2^K + 2$$

am numărat până la $K-1$.

$$= 2^K(2K-2-K)+2 = 2^K(K-2)+2$$

$$= (K-2)\cdot 2^K + 2 = K2^K - 2^{K+1} + 2$$



$\Rightarrow$ II) $-\log_2 \sum_q 2^q$

unde $K \sim \log n \Rightarrow$

$-\log 2(\log n \cdot n - 2n + 2)$

I) $\log n(2^K - 1) = (n-1)\log n$

(I) + (II) (+) $\rightarrow$ o.k

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

$T(n) \sim n \log n$

pb 2) Ce se întâmplă dacă OS select caută un nod ce nu e în arbore

↳ OS Select caută rangul unui element (și returnează ce găsește pe acea poziție)

cheie ce nu e în arbore $\Rightarrow i > size(x)$

se apelează OS_Select(right[x], i-r)

⇝ când depășesc dimensiunea arborelui

o să am $x \leftarrow right[x] = null$ deci

la căutarea dimensiunii o să am dim[left[null]+1]

⇓

eroare

⇓

evitare:

if $x \neq NIL$

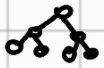
#3 BST to DLL

Analiză complexitate → pentru arbore echilibrat → best

Frunze: leftmost/rightmost ⇒ 0 ⇒ $\frac{n}{2}$ frunze

De câte ori? ⇓

parent[leaf] ⇒ (max) 1 ⇒ 1 ⇒ $\left[\frac{n}{2^2}, \frac{n}{2}\right)$

 min: 2 frunze → 1 părinte ⇒ $\left(\frac{n}{2}\right) : 2 = \frac{n}{2^2}$

max: 1 frunză → 1 părinte ⇒ $\frac{n}{2} - 1$ ↳ root

parent[parent[leaf]] ⇒ (max) 2 → $\left[\frac{n}{2^3}, \frac{n}{2^2}\right)$

⋮

↳ root ⇒ (max) h ⇒ 1 ⇔

$\frac{n}{2^2} \leq x_1 < \frac{n}{2}$

$\frac{n}{2^3} \leq x_2 < \frac{n}{2^2} \;/\cdot 2$

$\frac{n}{2^4} \leq x_3 < \frac{n}{2^3} \;/\cdot 3$

⋮

$\frac{n}{2^{h+1}} \leq x_h < \frac{n}{2^h}$ Dar $x_h = 1 \Rightarrow \frac{n}{2^h} = 1 \Rightarrow h = \log n$

⇔ $T(n) \to \sum_{k=1}^{\log n} k \frac{n}{2^k} = n \sum \frac{k}{2^k}$

$n \to \infty \Rightarrow \sum \frac{k}{2^k} \longrightarrow 2$
$(\log n \to \infty)$

⇔ rightmost, leftmost → $O(n)$

⇔ pt BST to DLL ⇒ parcurgerea → $O(n)$ (+)

$O(n)$

---

worst?

⇒ rightmost/left most ⇒
$0 + 1 + 2 + 3 + \ldots + n-1 = \frac{n(n-1)}{2}$

⇔ În total $O(n^2)$