# Tema II- structuri de date

Mîndruleanu Matei Daniel -143.

1. **Demonstrati ca un arbore binar care nu este plin nu poate corespunde unui cod optim. Pentru definitia unui cod optim, va rog sa consultati cursul despre coduri Huffman. Va reamintesc ca intr-un arbore binar plin, orice nod cu exceptia frunzelor are exact 2 fii.**

arbore binar plin = orice nod cu exceptia frunzelor are exact 2 fii

arbore optim = are costul minim

cost(arbore) = $\sum_{i \in frunza} \text{frecventa(i)} * adancime(i)$

presupun “T” => arbore neplin prin absurd optim

T neplin => (∃) nodul u ∈ T care are un singur fiu.

Daca elimin u din T, adancimea tuturor nodurilor din subarborele lui u va scadea =>

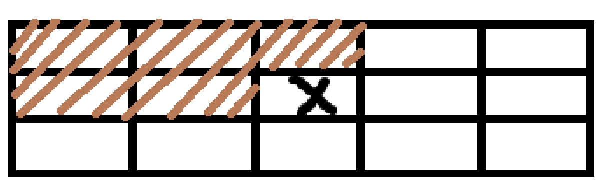
=> cost(T - u) < cost T => presupunere falsa => T nu este optim.

2. **Explicati cum se poate modifica metoda de sortare quicksort pentru ca aceasta sa ruleze in cazul cel mai defavorabil (i.e., worst-case) in timp O(n log n), presupunand ca toate numerele ce trebuie sortate sunt distincte.**

Quicksort merge in worst-case O($n^2$) cand alegem ca pivot primul numar, ultimul numar sau un numar random – care poate pica intr-una dintre extremitatile din array. Daca am alege pivotul in mijlocul array-ului sortat algoritmul s-ar descurca mai bine.

Putem alege pivotul folosind algoritmul de gasire al medianei intr-un vector ce merge in timp liniar. ( Impartim array-ul in vectori de cate 5 elemente in mod recurent. Calculam mediana feicareui vector. Adaugam toate medianele intr-un nou vector(daca dimensiunea noului vector este mai mare de 5 repetam pasul anterior) si ii calculam si acestuia mediana. ) Noua mediana este numarul dupa care ne ghidam.

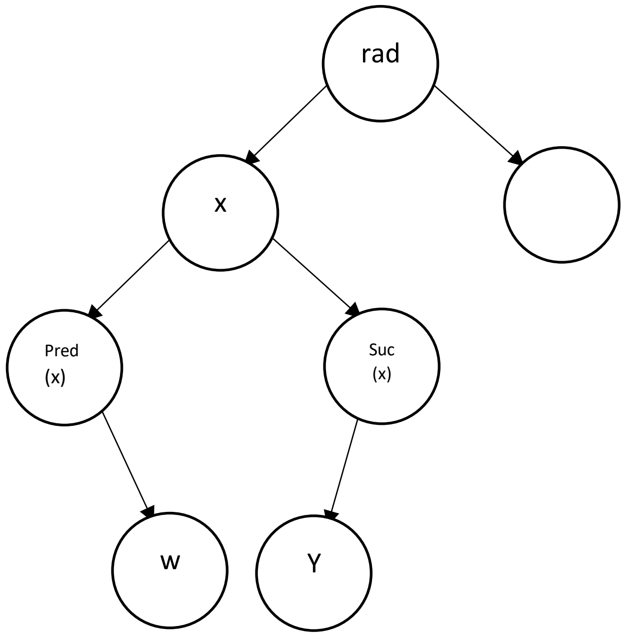
Daca alegem pivotul ca fiind mijlocul vectorului(mediana mentionata mai sus) operatia de recurenta devinde : T(n)=2T($\frac{n}{2}$)+O(n), care conform Teoremei Master este egal cu O(nlogn).



**3. Fie T un arbore binar de cautare si x un nod din arbore care are doi copii. Demonstrati ca succesorul nodului x nu are fiu stang, iar predecesorul lui x nu are fiu drept.**

Presupunem prin absurd ca y este fiul stang al succesorului lui x intr-un arbore binar de cautare T. Din proprietetile arborelui binar de cautare => suc(x) > x, y(fiul stang) < suc(x), dar din moment ce y se afla in partea dreapta a lui x => y > x -> contradictie, y = suc(x).
=>succesorul nodului x nu are fiu stang.

Presupunem prin absurd ca w este fiul drept al predecesorului lui x intr-un arbore binar de cautare T.
Din proprietatule arborelui binary de cautare => pred(x) < x, w(fiul drept) < pred(x), dar din moment ce w se afla in partea dreapta a lui x => w > x -> contradictie, w = pred(x).
=>predecesorului nodului x nu are fiu drept.




**4. Rezolvati recurenta T(n) = T(n/2) + T(n/3) + n. Demonstrati.**

$$\boldsymbol{T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{3}\right) + n}$$

$$\boldsymbol{T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{6}\right) + \frac{n}{2} + T\left(\frac{n}{6}\right) + T\left(\frac{n}{9}\right) + \frac{n}{3} + n}$$

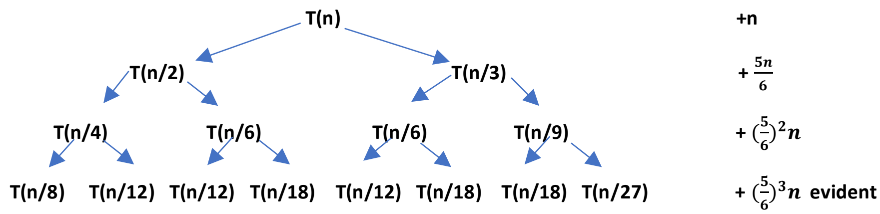
$$\boldsymbol{T(n) = T\left(\frac{n}{4}\right) + 2T\left(\frac{n}{6}\right) + \frac{5n}{6} + T\left(\frac{n}{9}\right) + n}$$

$$\boldsymbol{T(n) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{12}\right) + \frac{n}{4} + 2T\left(\frac{n}{12}\right) + 2T\left(\frac{n}{18}\right) + \frac{n}{3} + \frac{5n}{6} + T\left(\frac{n}{18}\right) + T\left(\frac{n}{27}\right) + \frac{n}{9} + n}$$

$$\boldsymbol{T(n) = n + \frac{5n}{6} + \frac{n}{4} + \frac{n}{9} + \frac{n}{3} + T\left(\frac{n}{8}\right) + 3T\left(\frac{n}{12}\right) + 3T\left(\frac{n}{18}\right) + T\left(\frac{n}{27}\right)}$$

$$T(n) = n + \frac{5n}{6} + \frac{25n}{36} + T\left(\frac{n}{8}\right) + 3T\left(\frac{n}{12}\right) + 3T\left(\frac{n}{18}\right) + T\left(\frac{n}{27}\right)$$

$$T(n) = n + \frac{5}{6}n + (\frac{5}{6})^2 n + T\left(\frac{n}{8}\right) + 3T\left(\frac{n}{12}\right) + 3T\left(\frac{n}{18}\right) + T\left(\frac{n}{27}\right)$$




Adancime : in cel mai rau caz O(n)

Frunze : cel mai mult $2^{log(n)}$=n; cost computational nT(1) = O(n)

Complexitate : n + $\frac{5}{6}n + (\frac{5}{6})^2 n$ +… ≤ $\sum_{k=0}^{\infty} (\frac{5}{6})^k = 6n$ = Θ(n)

=>total : Θ(n)

# Tema II – Structuri de Date

Bilbliografie


- https://edutechlearners.com/download/Introduction_to_algorithms-3rd%20Edition.pdf
- http://www.ccs.neu.edu/home/lieber/courses/algorithms/cs5800/sp14/help-sessions/slides/recurrences.pdf