Dima Oana-Teodora
Grupa 141

Tema 2 - Structuri de Date

2 puncte din oficiu

1 2 puncte

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 optim=arbore care are cost minim.
Rezolvare:

Fie T un arbore binar.
Presupun ca T este un arbore binar neplin. Acestea este un cod binar format din caractere ale alfabetului.

Cost-arbore $=\sum f(i)*adancime_i$, unde i este frunza
si f(i)=frecventa lui i in cod

Fie n nodul cu cea mai mare adancine din T, care are exact un fiu.
Daca n=radacina arbore=> n poate fi eliminat, iar adancimea fiecarui nod din subarborele determinat de n scade cu 1=> T reprezinta alfabetul cu un cost mai mic=> $cost_{(T-n)}<cost_T$ astfel ca, arborele nu poate fi Huffman, iar codul original nu e optim

---

2 2 puncte

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(nlogn), presupunand ca toate numerele ce trebuie sortate sunt distincte.

Rezolvare:
“Worst case scenario” pentru QuickSort are o complexitate de $O(n^2)$, apare atunci cand pivotul este ales un element extrem (maxim/minim), avand loc cand tabloul este dat sortat invers, iar primul element este ales drept pivot.

Dar, exista posibilitatea ca “worst case” sa fie redus la o complexitate de O(nlogn). Solutia are la baza faptul ca, elementul din mijloc al unui sir nesortat poate fi gasit in timp linear.Astfel ca, primul pas este gasirea medianei, apoi se imparte tabloul in jurul ei.

Algorim pesudocod -> am nevoie de 3 functii ajutatoare:

1)functie mediana(vector, n) :
-> returneaza mediana unui vector
2)functie partitionare(sir, l, r, x)
->il caut pe x si-l mut la final
->iterez prin sir, ii gasesc pozitia si interschimb
3)functie minim(sir, l, r, k) :
->aflu elementul minim
->k numarul de elemente din sir
->impart sirul in grupuri de 5 =>[((n+4)/5)] grupuri
->caut “mediana medianelor” unde ma folosesc de apeluri recursive la functie
->calculez pivotul folosind functia de partitionare

```
functie quickSort(sir, l, h) :unde l initial este 0 si h este n-1, n=dimensiune sir
    daca l < h atunci:
        //calculez dimensiunea sirului curent
        dim = h-l+1;
        //calcul mediana
        med =minim(sir, l, h, n/2);
        // impart in functie de mediana
        p = partitionare(sir, l, h, med);
        quickSort(sir, l, p - 1);
        quickSort(sir, p + 1, h);
```

Mediana:
$T(n) = T(n/5) + T(7n/10 +6) + O(n)$, unde n numar natural si c numar real
$T(n) \leq cn/5 + c(7n/10+6) + O(n)$
$T(n) \leq cn/5 + c + 7c/10 + 6c + O(n)$
$T(n) \leq 9cn/10 +7c + O(n)$
$T(n) \leq cn$=> algoritmul de gasire a medianei are timp de rulare “worst case” liniar.
Relatia de recurenta devine: $T(n)=2T(n/2)+O(n)$
Aplic Th. Master:

$T(n)=aT(n/b)+O(n^d)$, unde $a>0$, $b>1$ si $d\geq 0$

$T(n)=2T(n/2)+O(n) \Rightarrow a=2$ si $b=2 \Rightarrow \log_b a=1$ (1)

$n^d=n \Rightarrow d=1$ (2)

Din (1)+(2)$\Rightarrow d=\log_b a \Rightarrow T(n)=O(n\log n)$

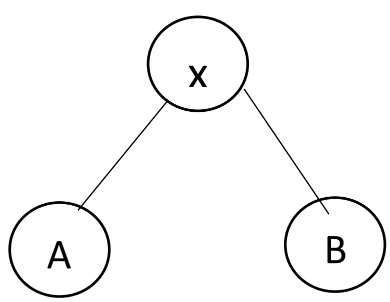
---

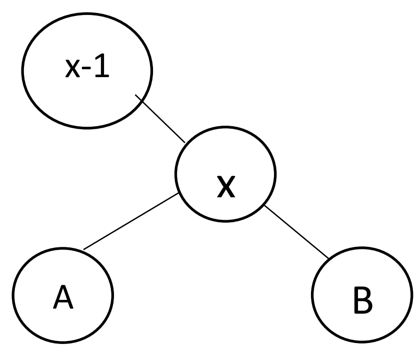
3 2 puncte

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.

Rezolvare:

Arbore binar de cautare => stanga<parinte<dreapta





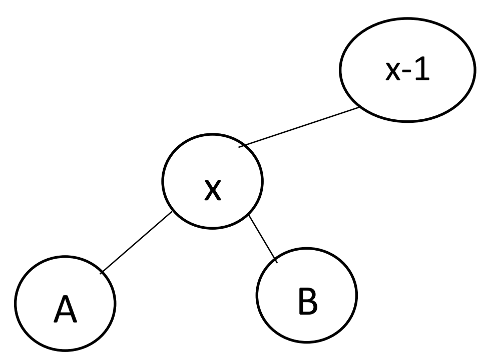
1)nodul predecessor(x-1)

x>x-1(Adevarat), dar A<B si A≠x-1

A<x-1, dar A>x-1 contradictie

2) x<x-1 (Fals)

Analog pentru nodul successor x+1



Cum x are 2 copii=> pot fi 2 subarbori A si B=> din demonstratia anterioara => nodul x-1 apartine ramurilor lui x

=>1) x-1 apartine B=>x-1>x(Fals)

2)x-1 aparitine A=> x-1<x (Adevarat)

Presupunem ca nodul x-1 are fiu drept pe nodul c => c>x-1 si nu exista nod intre x-1 si x => c>x-1=>c>x, dar c apartine A => c<x=> contradictie

Analog pentru x+1.

=>nodul successor/predecessor se afla doar in ramurile nodului x, dar nu neaparat sunt copii directi ai lui x.

---

4 2 puncte

Rezolvati recurenta T(n) = T(n/2) + T(n/3) + 1. Demonstrati.

Rezolvare:

$2T(n/3)+1 \leq T(n) \leq 2T(n/2)+1 \Rightarrow T_3(n) \leq T_1(n) \leq T_2(n)$

1. $T(n)=2T(n/2)+1$

Fie :$n=2^N$

$T(2^N)=2T(2^{N-1})+1$

$T(2^{N-1})=2T(2^{N-2})+1 \mid 2$

………………………..

$T(2^2)=2T(2)+1 \mid 2^{N-2}$

$T(2)=2T(1)+1 \mid 2^{N-1}$

Prin adunare membru cu membru or sa se reduca si o sa ramana:

$T(2^N)=2^N T(1)+(1+\ldots\ldots+2^{N-1})=2^N T(1)+(2^N-1 \rightarrow n-1)$

$n=2^N \Rightarrow \log_2 n=N \Rightarrow T(n)=nT(1)+n-1=2n-1$

2.Analog pentru $T3(n)=nT(1)+n-1=2n-1$ $(n=3^N)$

$T_3(n) \leq T_1(n) \leq T_2(n)$

nT(1)+n-1(Cleste)

Aplic Th. Master:

$T(n)=aT(n/b)+\Theta(n^d)$, unde $a>0$, $b>1$ si $d\geq 0$

$T(n)=2T(n/2)+1 \Rightarrow a=2$ si $b=2 \Rightarrow \log_b a=1$ (1)

$\Theta(n^d)=1 \Rightarrow n^d=n^0 \Rightarrow d=0$ (2)

Din (1)+(2)=> $d<\log_b a \Rightarrow T(n)=\Theta(n^{\log_2 2})=\Theta(n)$

$T(n)=2T(n/3)+1$

$\Theta(n^d)=1 \Rightarrow d=0$

$a=2$ si $b=3 \Rightarrow \log_3 2>\log_3 1=0 \Rightarrow \log_3 2>0 \Rightarrow \log_b a>d \Rightarrow T(n)=\Theta(n^{\log_3 2})$

$\Theta(n^{\log_3 2})\leq T(n)\leq \Theta(n)$

$n^{\log_3 2}=n^{0,63}>n^{0,5}=n^{1/2}=\sqrt{n} \Rightarrow \Theta(\sqrt{n})\leq T(n)\leq \Theta(n)$

$O(n)$

*Adaug aici si rezolvarea pentru exercitiul cu +n pentru ca apucasem sa-l fac pe cel cu +1 si am zis ca e pacat sa-l sterg ☺

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

Demonstrez ca :$T(n)=O(n)$

$T(n)<cn$; n numar natural nenul si c numar real nenul

Presupun ca $T(n/2)\leq cn/2$ si $T(n/3)\leq cn/3$

$T(n/2)+T(n/3)\leq cn/2 + cn/3 \mid +n \Rightarrow T(n) \leq cn/2 + cn/3+n \Rightarrow T(n)\leq(3cn+2cn+6n)/6 \Rightarrow$

$(3cn+2cn+6n)/6\leq cn \Rightarrow (n(5c+6))/6\leq cn \mid: n$ (diferit de 0)

$((5c+6))/6\leq c \mid 6 \Rightarrow 5c+6\leq 6c \Rightarrow -c\leq -6 \Rightarrow c\geq 6 \Rightarrow T(n)=O(n)$ are loc pentru $c>6$