

Structuri de date

Laboratorul 8

Mihai Nan

mihai.nan.cti@gmail.com

Grupa 315CC



Facultatea de Automatica și Calculatoare
Universitatea Politehnica din București
Anul universitar 2015 - 2016

1 Probleme de laborator

Observatie

In arhiva laboratorului, gasiti un schelet de cod de la care puteti porni implementarea functiilor propuse, avand posibilitatea de a le testa functionalitatea.

1.1 Probleme standard

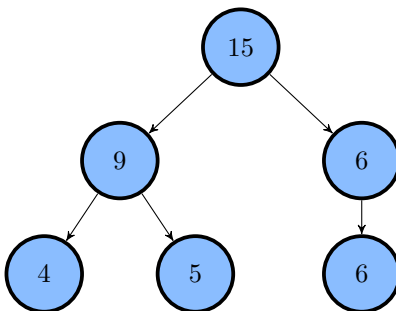
Problema 1 - 1 punct

Scrieti o functie care primeste ca parametru un arbore binar si determina numarul de frunze din arbore.

Problema 2 - 2 puncte

Scrieti o functie care primeste ca parametru un arbore binar si verifica daca indeplineste urmatoarea conditie: pentru fiecare nod intern ce nu este o frunza, valoarea nodului este egala cu suma valorilor celor doi copii. Vom considera valoarea 0 pentru copii ce sunt *NULL*.

Exemplu



Problema 3 - 3 puncte

Implementati o functie care primeste un arbore binar si doua numere ce reprezinta cheile a doua noduri din arbore. Functia va determina daca nodurile, care contin aceste chei, se afla pe acelasi nivel, dar nu au acelasi parinte.

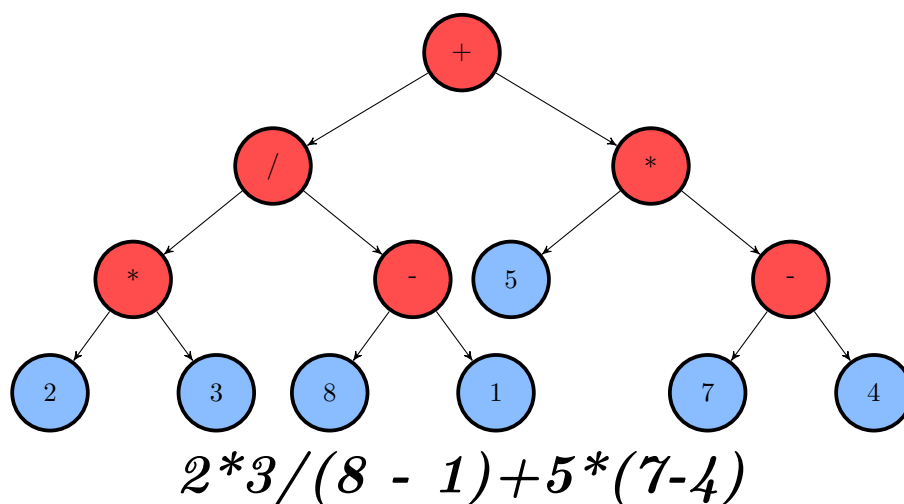
Problema 4 - 4 puncte

In matematica, o expresie aritmetica este reprezentata printr-un sir de caractere compus din: variabile, constante, operatori si paranteze. O astfel de expresie aritmetica poate fi reprezentata printr-un arbore binar. Nodul radacina

contine un operator, iar fiecare din subarborii sai reprezinta fie numele unei variabile sau unei constante, fie o alta expresie. in arborele binar asociat unei expresii nu apar parantezele. Pentru operatiile comutative, ordinea subarborilor stang si drept nu conteaza, acestia pot fi inversati, iar arborele este bine construit. in cazul operatiilor necomutative, ordinea subarborilor este strict precizata, adica subarborii nu pot fi inversati in reprezentare. in general, arborele binar asociat unei expresii nu este unic.

Construiti arborele sintactic pentru o expresie data in forma infixata si evaluati expresia pe baza acestei reprezentari.

Exemplu



1.2 Problema bonus

Problema 5 - 2 puncte

Scrieti o functie care determina distanta dintre doua chei dintr-un arbore binar. Distanța dintre doua noduri este numarul minim de noduri traversate pentru a ajunge de la un nod la altul.

2 Interviu

Observatie

Aceasta sectiune este una optionala si incearca sa va familiarizeze cu o serie de intrebari ce pot fi adresate in cadrul unui interviu tehnic. De asemenea, aceasta sectiune poate fi utila si in pregatirea pentru examenul final de la aceasta disciplina.



Intrebari interviu

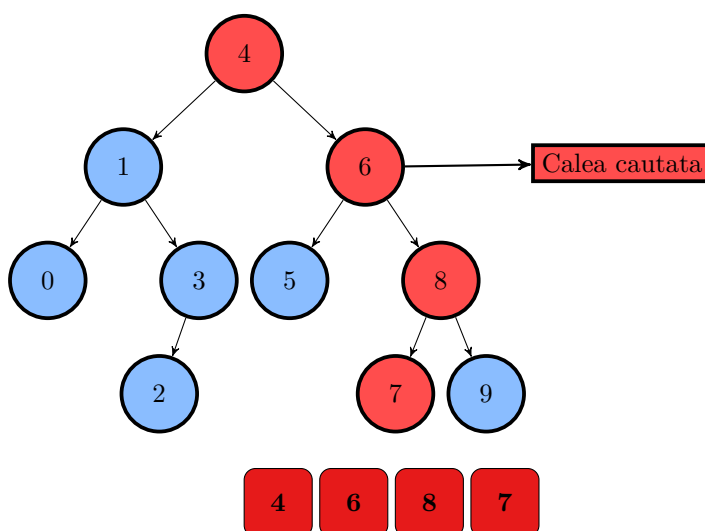
1. Consideram un arbore de cautare initial vid. Desenati arborele de cautare obtinut prin inserarea pe rand a cheilor $P, R, O, B, L, E, M, A, F, O, A, R, T, E, U, S, O, A, R, A$.
2. Consideram un arbore binar reprezentat prin vectorii $St = (2, 4, 0, 5, 0, 0)$, $Dr = (3, 0, 0, 6, 0, 0)$, in care radacina este nodul cu cheia 1. Cate frunze are acest arbore? Ce puteti spune despre acest arbore?
3. Intr-un arbore binar cu 9 varfuri, in care toate varfurile ce nu sunt frunze au exact 2 fii, care este adancimea maxima?
4. Cu cat este egala lungimea prefixului in cazul unui arbore de regasire?
5. Ce avantaje prezinta un arbore, comparativ cu restul structurilor de date studiate?
6. Ce formule exista intre numarul N de noduri al unui arbore binar si inaltimea sa H ?
7. Ce reprezinta succesorul in inordine al unui nod intr-un arbore binar de cautare?
8. Ce se intampla daca intr-un arbore binar de cautare introducem un sir, de numere intregi, sortat crescator? Dar in cazul unui sir sortat descrescator? Argumentati!

2.1 Probleme interviu

Problema 1: Implementati o functie care verifica daca intr-un arbore binar de cautare exista o cale compusa din noduri ale caror chei au suma **K**. In cazul in care exista o astfel de cale de la radacina la oricare nod al arborelui, afisati cheile nodurilor care compun aceasta cale, in ordinea in care acestea apar in reprezentarea arborelui.

Exemplu

K = 25



Problema 2: Sa se implementeze o functie pentru inserarea unui nod intr-un arbore binar de cautare, fara a utiliza recursivitatea.

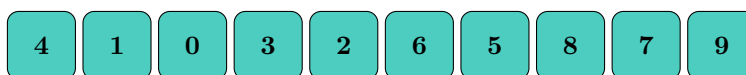
Problema 3: Implementati un program care sa construiasca un arbore binar de cautare pornind de la sirurile rezultate in urma parcurgerii sale in preordine si inordine.

Exemplu

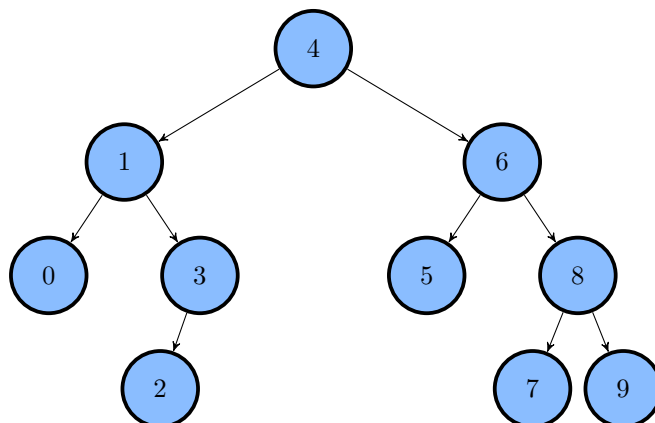
Parcuregere in inordine



Parcuregere in preordine



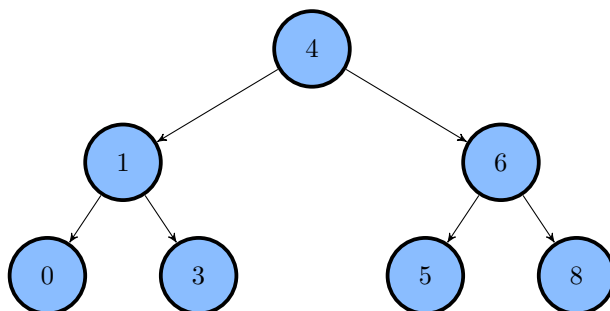
Arborele rezultat



Problema 4: Dandu-se un arbore binar de cautare si valoarea unei chei a unui nod, implementati o functie care determina toti stramosii acelui nod din arbore. In cazul in care nodul este chiar radacina arborelui, se va afisa mesajul *Radacina arborelui*.

Problema 5: Implementati o functie ce realizeaza parcurgerea unui arbore binar de cautare in **Zig - Zag**.

Exemplu

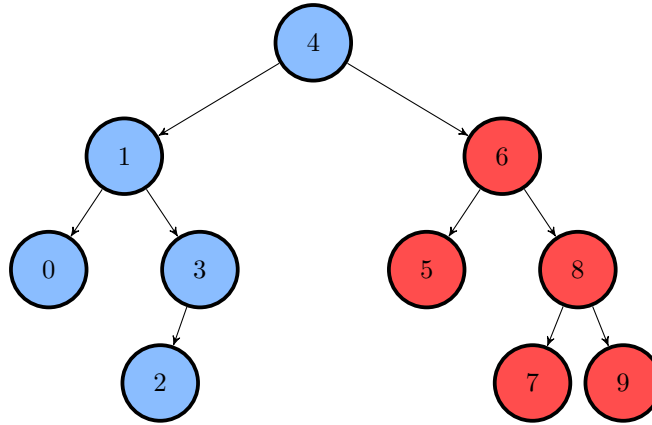


Parcurgerea in Zig - Zag



Problema 6: Definiti o functie care primeste ca argument un arbore binar de cautare si determina subarborele maximal al acestuia, ca numar de noduri. Pentru a se testa functionalitatea acestei functii, se va afisa parcurgerea in postordine a subarborelui gasit. In cazul in care exista doi subarbori, avand acelasi numar de noduri in componenta, se va alege subarborele de suma maxima.

Exemplu



Problema 7: Implementati o functie care imbină doi arbori binari de cautare, afișând parcurgerea în ordine a arborelui binar rezultat prin imbinarea acestora.

⚠ IMPORTANT !



Această funcție se poate utiliza în cazul operației de ștergere a rădăcinii unui arbore binar de căutare.

Feedback

Pentru îmbunătățirea constantă a acestui laborator, vă rog să completați formularul de feedback disponibil aici.

De asemenea, vă rog să semnalati orice greșeală / neclaritate depistată în laborator pentru a o corecta.

Vă mulțumesc anticipat!