

Dokumentacija

1. laboratorijska vježba iz Naprednih algoritama i struktura podataka

Student: Lea Suć

Matični broj: 0036444175

Zadatak: Implementacija AVL stabla s mogućnošću brisanja čvorova

Službeni zadatak

Napisati program koji učitava niz prirodnih brojeva iz ASCII datoteke (po pretpostavci, datoteka nije prazna) i upisuje ih u (inicijalno prazno) AVL stablo istim redoslijedom kao u datoteci. Program može biti konzolni ili s grafičkim sučeljem, po vlastitom izboru. Konzolni program naziv ulazne datoteke treba primiti prilikom pokretanja kao (jedini) argument s komandne linije, a grafički iz odgovarajućeg sučelja po pokretanju programa. Nakon upisa svih podataka, ispisati izgrađeno stablo na standardni izlaz (monitor). Program zatim treba omogućiti dodavanje novih čvorova te nakon svake promjene treba ponovo ispisati stablo. Datoteka treba biti tekstualna, a brojevi u datoteci odvojeni po jednim razmakom (space). Kraj datoteke treba biti označen standardnim EOF znakom. Treba dodati i brisanje čvora iz stabla.

1. AVL stablo

AVL stablo je binarno stablo pretraživanja koje zadovoljava pravilo da faktori ravnoteže svih čvorova moraju biti -1, 0 ili 1, pri čemu se faktor ravnoteže računa uobičajeno kao razlika visina desnog i lijevog podstabla. AVL algoritam uravnotežuje stablo lokalno, ali ne garantira savršenu uravnoteženost cijelog stabla. Dodavanje čvora u AVL stablo sastoji se od pronalaženja mjesta na koje se čvor ubacuje i uravnotežavanja stabla. Faktori ravnoteže osvježavaju od roditelja dodanog čvora prema korijenu, a ukoliko osvježeni faktor ravnoteže nekog čvora postane 2 ili -2, primjenjuju se jednostruke ili dvostruke rotacije, ovisno o pojedinom slučaju. Kod brisanja se najprije uklanja čvor pomoću brisanja kopiranjem, a potom na isti način uravnotežava stablo od roditelja čvora koji je uklonjen prema korijenu, uzimajući pritom u obzir dodatne moguće slučajeve.

2. Programsko ostvarenje

Rješenje zadatka ostvareno je u programskom jeziku Java. U izradi je korišteno integrirano razvojno okruženje Eclipse IDE za Javu. Sučelje je konzolno.

Čvorovi stabla predstavljeni su klasom *Node* i sadrže podatak o pohranjenoj vrijednosti (`private int data`), referencu na roditelja (`private Node parent`), te reference na lijevo i desno dijete (`private Node leftChild`, `private Node rightChild`). Klasa *Node* sadrži metodu koja vraća faktor ravnoteže čvora, računajući razliku visina desnog i lijevog djeteta. Metoda `getHeight()` vraća visinu čvora, odnosno broj razina ispod čvora, na način da gleda visine lijevog i desnog djeteta, te uzme veću od njih i pribroji joj 1.

Sâmo AVL stablo implementirano je u klasi *AVLTree*, koja ima jednu člansku varijablu, `private Node rootNode`, koja predstavlja korijen stabla. Stablo se gradi dodavanjem novih čvorova. Ova klasa sadrži metode za pretraživanje, ubacivanje, brisanje, uravnotežavanje i ispis stabla, koje će biti detaljnije opisane u nastavku, te mnogobrojne pomoćne metode, poput metode za jednostruku ili dvostruku rotaciju.

Klasa *MainClass* sadrži metodu `main` i vrši učitavanje AVL stabla iz tekstualne datoteke, poziva metodu za ispis, te metode za dodavanje ili brisanje čvorova stabla, u ovisnosti o unosu korisnika.

2.1. Dodavanje čvora u AVL stablo

Novi čvor se dodaje pomoću metode `public void insert (int value)`, tako da se pronade mjesto na koje se čvor ubacuje kao i u običnom binarnom stablu (za svaki čvor su čvorovi s vrijednostima manjim od vrijednosti koju on sadrži lijevo od njega, a s većim desno od njega). Nakon što se stvori i ubaci novi čvor koji sadrži vrijednost `value`, poziva se metoda koja vrši uravnotežavanje stabla. Ukoliko već postoji čvor sa zadanom vrijednošću, ispisat će se odgovarajuća poruka, a čvor se neće dodati u stablo.

2.2. Brisanje čvora u AVL stablu

U svrhu brisanja čvora iz stabla poziva se metoda `public boolean delete (int value)`. Najprije se pretražuje stablo pomoću metode `public Node search (int value)`, koja vraća referencu na pronađeni element, a ukoliko on ne postoji vraća `null`. Ako stablo ne sadrži traženi

element, ispisuje se odgovarajuća poruka, a u suprotnom se ispituju različiti slučajevi s obzirom na koje se čvor briše.

Ako se briše čvor koji je jedini u stablu, AVL stablo postaje prazno i ispisuje se odgovarajuća poruka. Ako je čvor list, on se jednostavno obriše. Ako se briše čvor koji ima samo jedno dijete i nije korijen, referenca njegovog roditelja se usmjeri na njegovo dijete. Inače, ako se briše korijen koji ima jedno dijete, njegovo dijete postaje novi korijen stabla. Složeniji slučaj je kod brisanja čvora s dva djeteta, jer je tada potrebno pozvati metodu koja briše čvor kopiranjem, `public Node deleteByCopying (Node node)`. Nakon svakog brisanja poziva se metoda koja uravnotežava AVL stablo od roditelja obrisanih čvora prema korijenu.

2.3. Uravnotežavanje AVL stabla

Uravnotežavanje stabla odvija se pomoću metode `private void balanceTree (Node node)`, koja je prikazana na slici 1.

```
private void balanceTree (Node node){
    while (node != null){
        int nodeFactor = node.getBalanceFactor();
        if (nodeFactor == 2){
            int childFactor = node.getRightChild().getBalanceFactor();
            if (childFactor == 1 || childFactor == 0){
                singleRotate(node, Side.LEFT);
            }
            else if (childFactor == -1){
                doubleRotate(node.getRightChild(), Side.RIGHT);
            }
        }
        else if (nodeFactor == -2){
            int childFactor = node.getLeftChild().getBalanceFactor();
            if (childFactor == 1){
                doubleRotate(node.getLeftChild(), Side.LEFT);
            }
            else if (childFactor == -1 || childFactor == 0){
                singleRotate(node, Side.RIGHT);
            }
        }
        node = node.getParent();
    }
}
```

Slika 1. Metoda koja uravnotežava stablo pozivajući metode koje vrše jednostruku ili dvostruku rotaciju

Ova metoda prima referencu na čvor od kojeg potom osvježava faktore ravnoteže prema korijenu. Kad naiđe na čvor s faktorom ravnoteže 2 ili -2, znači da je tom čvoru narušena ravnoteža zbog dodavanja ili brisanja čvorova, i potrebno je intervenirati izvodeći običnu ili dvostruku rotaciju. Tada se provjerava faktor ravnoteže njegovog djeteta (ako je roditeljev faktor ravnoteže pozitivan desnog djeteta, inače lijevog), i ako su faktori ravnoteže djeteta i roditelja istog predznaka (2 i 1 ili -2 i -1), poziva se metoda koja rotira dijete oko roditelja (ovisno o predznaku roditelja obavlja se lijeva ili desna rotacija), a ako su suprotnog predznaka, poziva se metoda za dvostruku rotaciju, koja najprije rotira djetetovo dijete oko djeteta, pa potom novo dijete oko roditelja. Jednostruka rotacija se provodi i kad je djetetov faktor ravnoteže jednak nuli, što se može dogoditi samo prilikom brisanja.

3. Primjer rada programa

Primjer izvršavanja programa kod učitavanja čvorova stabla iz tekstualne datoteke koja sadrži vrijednosti 2 i 5 odvojene razmakom, te unos čvora 6 i potom brisanje čvora 5 prikazan je na slici 2.

```

@ Javadoc Declaration Console
MainClass (1) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (22.10.2011. 18:01:16)
Ucitavanje iz datoteke: C:\AVLTree.txt

~
5
~
2
~

Unesite 'u' za umetanje u stablo, 'b' za brisanje cvora iz stabla ili 'q' za izlaz.
u
6

~
6
~
5
~
2
~

b
5

~
6
~
2
~

```

Slika 2. AVL stablo ispisano je rotirano ulijevo za 90 stupnjeva, korijen je u početku 2 a desno dijete 5. Znak „~“ predstavlja null.

4. Zaključak

Algoritam za konstrukciju AVL stabla idejno je vrlo jednostavan. Programska implementacija također nije jako zahtjevna, premda je potrebno paziti na mnogo mogućih situacija odjednom, posebno kod brisanja i podešavanja ravnoteže nakon promjena.

Priloženo programsko rješenje moguće je poboljšati, primjerice tako da se umjesto rekurzivnog računanja visine čvora radi s članskom varijablom `height`, koja bi se dodala u klasu *Node*. Isto tako moguća je implementacija bez reference na roditelja u pojedinom čvoru, no takvo rješenje bilo bi znatno složenije.