

Fakultet elektrotehnike i računarstva
Zavod za primjenjeno računarstvo

Napredni algoritmi i strukture podataka

1. laboratorijska vježba

Antonia Elek, 0036477925

Zagreb, 5.11.2016.

1. Zadatak

Zadatak za 17 bodova

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. Dodati i brisanje čvora iz stabla.

Napomena: datoteka treba biti tekstualna, a brojevi u datoteci odvojeni po jednim razmakom (space). Kraj datoteke treba biti označen standardnim EOF znakom.

2. Rješenje zadatka

2.1. Teorijski uvod

AVL stablo je binarno stablo koje zadovoljava AVL definicijsko pravilo: faktori ravnoteže svih čvorova moraju biti -1 , 0 ili 1 . Pritom se faktor ravnoteže čvora računa kao razlika visina desnog i lijevog podstabla.^[1]

2.2. Implementacija

Rješenje zadatka ostvareno je u programskom jeziku C#.

Zadatak se može razdijeliti na nekoliko manjih problema:

- Struktura čvora
- Dodavanje čvora u stablo
- Uravnoteživanje stabla nakon dodavanja čvora
- Brisanje čvora iz stabla
- Uravnoteživanje stabla nakon brisanja čvora
- Desna rotacija
- Lijeva rotacija

2.2.1. Struktura čvora

Svaki čvor sadrži svoju vrijednost (integer), visinu, reference na lijevo i desno djetete, faktor ravnoteže koji se računa kao razlika visina desnog i lijevog djeteta.

2.2.2. Dodavanje čvora u stablo

Dodavanje čvorova u AVL stablu ostvareno je kao dodavanje u binarno stablo za pretraživanje. Od korijena stabla se ide prema listovima i za svaki čvor na putu provjerava da li je novi podatak veći ili manji od podatak u tom čvoru. Ako

je manji nastavljamo lijevim podstablom, a ako je veći, desnim. Ako čvor nema podstablo kojim bi se put nastavio, stvara se novi čvor. Na kraju se osvježavaju visine čvorova na putanji kojom se prošlo prilikom dodavanja novog čvora.

Dodavanje čvora - pseudokod:

```
InsertInto(key, root):
    if (root does not exist)
        return new node with key
    if (key < root.Key)
        InsertInto(key, root.Left)
    else
        InsertInto(key, root.Right)

    root.Height = 1 + Max(root.Left, root.Right)

    BalanceAfterInsert(root)
```

2.2.3. Uravnotežavanje stabla nakon dodavanja čvora

Nakon dodavanja novog čvora u stablo i osvježavanja faktora ravnoteže čvorova, može se dogoditi da je narušeno definicijsko pravilo AVL stabla. Potrebno je uravnotežiti stablo, a prvi i jedini čvor koji će zahtijevati intervenciju bit će onaj čvor koji će imati faktor ravnoteže 2 ili -2.

Uravnotežavanje stabla - pseudokod:

```
BalanceAfterInsert(node)
    if (node.Balance == -2 && node.Left.Balance == -1)
        RightRotate(node)

    if (node.Balance == 2 && node.Right.Balance == 1)
        LeftRotate(node)

    if (node.Balance == -2 && node.Left.Balance == 1)
        LeftRotate(node.Left)
        RightRotate(node)

    if (node.Balance == 2 && node.Right.Balance == -1)
        RightRotate(node.Right)
        LeftRotate(node)
```

2.2.4. Brisanje čvora iz stabla

Brisanje čvora u AVL stablu ostvareno je brisanjem kopiranjem. Stablo se pretražuje od korijena i kada je pronađen čvor kojeg je potrebno izbrisati, traži se zamjenski čvor - čvor sa najvećim manjim podatkom od čvora koji se briše. Podaci iz zamjenskog čvora prepišu se u čvor koji je označen za brisanje, a zamjenski čvor se uklanja. Na kraju se osvježavaju visine čvorova u stablu.

Brisanje čvora - pseudokod:

```
DeleteNode(key, root):
    if (key < root.Key)
        DeleteNode(root.Left, key)
```

```

else if (key > root.Key)
    DeleteNode(root.Right, key)
else
    if (root.Left does not exist OR root.Right does not exist)
        temp = root.Left
        if (temp does not exist)
            temp = root.Right
        root = temp
    else
        var temp = SmallestInSubtree(root.Right)
        root.Key = temp.Key
        DeleteNode(root.Right, temp.Key)

root.Height = Max(root.Left, (root.Right)) + 1

BalanceAfterDelete(root)

```

2.2.5. Uravnotežavanje stabla nakon brisanja čvora

Brisanjem čvora može doći do promjene visine stabla, što za posljedicu može imati narušavanje definicijskog pravila. Ako je nekom čvoru faktor ravnoteže postao veći od 1 ili manji od -1 stablo je nebalansirano i potrebno ga je uravnotežiti.

Uravnotežavanje stabla - pseudokod:

```

BalanceAfterDelete(node)
    if (node.Balance < -1 && node.Left.Balance <= 0)
        RightRotate(node)

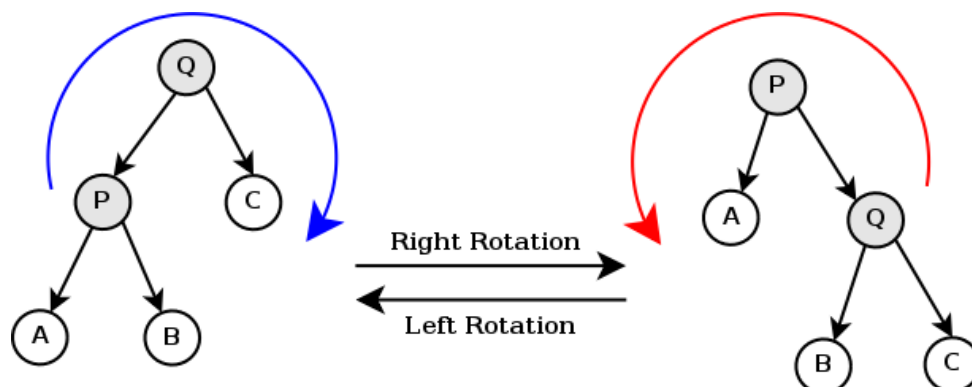
    if (node.Balance < -1 && node.Left.Balance > 0)
        LeftRotate(node.Left)
        RightRotate(node)

    if (node.Balance > 1 && node.Right.Balance >= 0)
        LeftRotate(node)

    if (node.Balance > 1 && node.Right.Balance < 0)
        RightRotate(node.Right)
        LeftRotate(node)

```

2.2.6. Rotacije



Slika 1. – Desna i lijeva rotacija u stablu [2]

Desna rotacija - pseudokod:

```
RightRotate(p) :  
    q = p.Left  
    qr = q.Right  
  
    q.Right = p  
    p.Left = qr  
  
    p.Height = Max(p.Left, p.Right) + 1  
    q.Height = Max(q.Left, q.Right) + 1
```

Lijeva rotacija - pseudokod:

```
LeftRotate(p) :  
    var q = p.Right  
    var ql = q.Left  
  
    q.Left = p  
    p.Right = ql  
  
    p.Height = Max(p.Left, p.Right) + 1  
    q.Height = Max(q.Left, q.Right) + 1
```

3. Zaključak

Izrađena je konzolna aplikacija koja iz datoteke učitava niz prirodnih brojeva iz datoteke, gradi AVL stablo i ispisuje ga na ekran te omogućuje dodavanje novih i brisanje postojećih čvorova. Implementirane su operacije dodavanja čvora u AVL stablo, brisanje čvora i ispis stabla s pripadajućim faktorima ravnoteže svih čvorova.

4. Literatura

- [1] Nikica Hlupić i Damir Kalpić, Napredni algoritmi i strukture podataka, FER, http://www.fer.unizg.hr/download/repository/Strukture_podataka.pdf, 1.11.2016.
- [2] Wikipedia, https://en.wikipedia.org/wiki/Tree_rotation, 1.11.2016.