

Fakultet elektrotehnike i računarstva
Zavod za primjenjeno računarstvo

Napredni algoritmi i strukture podataka

1. laboratorijska vježba

Ivan Štignedec, 0036493051

Zagreb, 9. studenog 2018.

1. Zadatak

Zadaci za 11 bodova – zadatak 1)

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.

2. Rješenje zadatka

2.1. Teorijski uvod

AVL stablo binarno je uravnoteženo stablo, koje ne jamči savršenu uravnoteženost, već je uravnoteženo lokalno kako bi se ubrzale operacije nad stablima, u usporedbi sa DSW algoritmom koji jamči savršenu uravnoteženost stabla. AVL stablo zadovoljava AVL definicijsko pravilo, koje nalaže da faktor ravnoteže svakog čvora u stablu mora biti -1, 0 ili 1. Pri tome se faktor ravnoteže računa kao razlika visine desnog i lijevog podstabla. Samo uravnotežavanje stabla potrebno je izvršiti nakon dodavanja novog ili brisanja postojećeg čvora stabla. To će biti moguće izvesti praćenjem i ažuriranjem faktora ravnoteže čvorova stabla.

2.2. Implementacija

Rješenje ovog problema implementirano je u programskom jeziku C++, koristeći razvojno okruženje MS Visual Studio 2015. Program je konzolni, te se pokreće uz jedan argument – naziv datoteke koja sadrži elemente čvorova koji će inicijalno biti dodani u stablo. Program će stvoriti uravnoteženo AVL stablo u kojega se mogu upisivati dodatni elementi (čvorovi), a izlazak iz programa upisom znaka 'x' unutar konzole.

2.2.1. Dodavanje novog čvora u stablo

Prije dodavanja novog čvora u stablo, stablo je potrebno inicijalizirati pozivanjem odgovarajućeg konstruktora. Također je potrebno stvoriti čvor korijena stabla, te tada iterativno pozivati funkciju za dodavanje čvora u stablo. Postoje dvije funkcije za dodavanje čvora u stablo, prva oblika "void dodaj(int element);" koju pozivamo iz glavne (main) funkcije prilikom dodavanja svakog čvora, unutar koje se postavlja novi čvor na mjesto korijena stabla, i druga, oblika "cvor* dodaj_cvor(cvor* cv, int broj);"

unutar koje se radi uravnotežavanje stabla. Iz funkcije “dodaj” poziva se funkcija “dodaj_cvor” koja, počevši od korijena stabla uspoređuje vrijednosti postojećih elemenata s vrijednošću novog elementa i rekurzivno poziva funkciju kako bi došla do mjesta na koje se umeće novi čvor. Tada se provjerava uvjet ravnoteže, te, ako on nije ispunjen, potrebno je izvršiti odgovarajuće rotacije kako bi se stablo uravnotežilo. Pseudokod uravnotežavanja AVL stabla prikazan je u nastavku.

Pseudokod za funkciju uravnotežavanja AVL stabla:

```
izračunaj FR roditelja

ako je FR roditelja == 2
    ako je FR čvora == 1
        rotiraj čvor ulijevo
        vrati čvor
    inače ako je FR čvora == -1
        rotiraj lijevo dijete udesno
        rotiraj lijevo dijete ulijevo
        vrati čvor

inače ako je FR roditelja == -2
    ako je FR čvora == -1
        rotiraj čvor udesno
        vrati čvor
    inače ako je FR čvora == 1
        rotiraj desno dijete ulijevo
        rotiraj desno dijete udesno
        vrati čvor

ponovi za roditelja
```

Iz pseudokoda je moguće izdvojiti dva slučaja (i dva zrcalna) rotacije čvorova. Ako se za FR roditelja dobije pozitivna (negativna) vrijednost, te ako je i FR djeteta pozitivan (negativan), tada to nazivamo izravnati slučaj u kojemu radimo samo jednu rotaciju oko čvora roditelja. Ako su predznaci različiti, dobiven je izlomljeni slučaj pri kojemu se rade dvije rotacije čvorova.

2.2.2. Rotacije čvorova

Lijeva i desna rotacija čvorova implementirane su pomoću pomoćnog čvora koji (opisujući desnu rotaciju) pokazuje na lijevo dijete trenutnog čvora dobivenog kao parametar funkcije. Trenutni (roditeljski) čvor pokazati će na desno dijete lijevog djeteta. Tada će desni pokazivač lijevog djeteta trenutnog čvora pokazati na trenutni čvor (roditelj). Funkcija će, ako lijevo dijete trenutnog čvora postoji (nije NULL) vratiti taj čvor, a ako ne postoji, vraća trenutni čvor (roditelj). Postupak je jednak i za lijevu rotaciju, uz zrcalnu zamjenu smjerova pokazivača.

2.2.3. Ispis AVL stabla

Postoje tri vrste ispisa AVL stabla, koje se pokreću prilikom inicijalizacije i nakon svakog umetanja novog čvora u stablo : inorder ispis, preorder ispis i obični ispis stabla. Prilikom inorder ispisa, prvo se ispisuje lijevo dijete, roditelj, pa desno dijete.

Kod preorder ispisa, prvo se ispisuje roditelj, pa lijevo i desno dijete. Obični ispis oblikuje stablo tako da ispiše vrijednost čvora roditelja, pa njegovo lijevo i desno dijete, u obliku "Čvor : <vrijednost>, Lijevo dijete : <vrijednost>, Desno dijete : <vrijednost>".

3. Zaključak

Prilikom implementacije dokazana je teorijska uspješnost korištenja ovog postupka za izgradnju uravnoteženog stabla, pri čemu osnovnu prednost u odnosu na algoritam DSW ima brzina i jednostavnost. Podaci u čvorovima bili su tipa "int", cijeli brojevi, a algoritam bi se mogao proširiti i na ostale tipove podataka kako bismo u širem kontekstu mogli koristiti ovaj pristup.

4. Literatura

[1] N. Hlupić, D. Kalpić, Napredni algoritmi i strukture podataka, Zagreb, FER, 2009.