

Ivan Herak
JMBAG: 0036423930

NAPREDNI ALGORITMI I STRUKTURE PODATAKA

Dokumentacija za 1. laboratorijsku vježbu

Listopad, 2012.

Zadatak (17 bodova):

Napisati program koji učitava niz prirodnih brojeva iz ASCII datoteke (po pretpostavci, datoteka nije prazna) i upisuje ih u (inicijalno prazno) crveno-crno (RB) 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) pri čemu treba nekako naznačiti boje čvorova. Program zatim treba omogućiti dodavanje novih čvorova te nakon svake promjene treba ponovo ispisati stablo. Potrebno je dodati i mogućnost brisanja čvora iz stabla.

Za implementaciju navedenog problema korišten je programski jezik C#. Svaki čvor definiran je klasom koja sadrži slijedeće podatke: vrijednost elementa čvora (value), pokazivač na lijevo i desno dijete čvora (Left, Right), te visinu čvora (height). Stablo je inicijalno prazno, te sadrži samo jedan čvor, odnosno korijen stabla. Sve metode (funkcije) koje se obavljaju nad AVL stablom definirane su unutar klase AVL.

Na početku rada otvara se sučelje iz kojeg je moguće, pomoću dugmeta „Odaber...“ odabrati ulaznu datoteku koja sadrži početne elemnte. Ti elementi se dodaju u stablo, te se na kraju na tekstualnom polju (textBox) ispisuje stablo. Nakon toga korisnik može dodati proizvoljan element u stablo, ili izbrisati postojeći. Nakon svake promjene stablo se nanovo ispisuje na tekstualnom polju.

Dodavanje novog elementa u stablo vrši se unutar funkcije *insert()*. Nakon pronalaska praznog mjesta u stablu formira se novi čvor i dodjeljuje mu se vrijednost novog elementa. Kretanje kroz stablo vrši se rekurzivno, od korijena prema listovima stabla. Nakon dodavanja novog čvora, započinje kretanje po čvorovima od roditelja novog čvora prema korijenu i izvodi se balansiranje stabla: provjerava se faktor ravnoteže čvora te se po potrebi izvodi jedan od četiri slučaja rotacije elemenata stabla u svrhu balasiranja trenutnog podstabla. Rotacije čvorova se vrše pomoću metoda *RotateRight*, *RotateLeft*, *RotateLeftRight* i *RotateRightLeft*, čija će funkcionalnost biti opisana kasnije. Nakon svakog pomaka kroz čvorove prema korijenu stabla osvježava se visina čvora funkcijom *setHeight*, i to tako da se pomoću funkcije *max()* odredi maksimum vrijednosti visine lijevog i desnog podstabla trenutnog čvora, te se trenutnom čvoru dodijeli ta vrijednost uvećana za jedan.

Brisanje elementa iz stabla izvodi se pomoću funkcije *delete*, i implementacija ove operacije nešto je složenija od dodavanja novog elementa. Ako je zadana vrijednost pronađena u stablu, postoji nekoliko mogućih scenarija brisanja čvora iz stabla:

- ako je i lijevo i desno podstablo čvora prazno, onda je čvor zapravo list stabla: čvor se briše iz stabla.
- ako čvor koji brišemo ima samo jedno podstablo, a drugo je prazno, tada se čvor briše kopiranjem (deletion by copying) desnog, odnosno lijevog djeteta (prvog čvora podstabla) na njegovo mjesto.
- ako čvor koji brišemo ima i lijevo i desno podstablo, tada se poziva funkcija *findMin*. Ova funkcija radi slijedeće: na mjesto obrisano čvora zapisuje se

vrijednost najmanjeg elementa njegovog desnog podstabla. Ako podstablo ima samo desnu granu, tada se vraća vrijednost prvog elementa desne grane. U protivnom, radi se pomak po lijevoj grani podstabla dok se ne pronađe najmanji element.

Nakon brisanja stabla, opet je potrebno napraviti balansiranje stabla. Balansiranje se vrši od roditelja zamjenskog čvora prema korijenu stabla. Za rotaciju čvorova koriste se iste funkcije koje su korištene kod umetanja novog elementa u stablo. Ovisno od faktora ravnoteže čvora i faktora ravnoteže njegovog desnog, odnosno lijevog djeteta, moguće je šest slučajeva u kojima je potrebna intervencija, odnosno rotacija čvorova radi uravnoteženja visina podstabla. Ovo se, kao i kod dodavanja elementa u stablo, izvršava rekursivno prema korijenu stabla. Nakon svakog pomaka kroz čvorove i eventualnih intervencija, osvježava se visina čvora u ovisnosti od visina lijevog i desnog podstabla.

Za implementaciju rotacija čvorova u svrhu balansiranja stabla, koriste se ranije spomenute funkcije *RotateRight*, *RotateLeft*, *RotateLeftRight* i *RotateRightLeft*. Funkcije *RotateLeft*, odnosno *RotateRight*, koriste se za izvedbu jednostruke rotacije lijevog, odnosno desnog čvora djeteta oko čvora roditelja. Unutar ovih funkcija koristi se pomoćni čvor (objekt tipa *Node*) u koji se zapisuju vrijednosti podstabla nakon izvedenih rotacija, te se pokazivač na njezin korijen zapisuje na mjesto korijena podstabla nad kojim je rotacija bila izvedena. Funkcije *RotateLeftRight* i *RotateRightLeft* izvede dvostruku rotaciju čvorova, i to tako da se prvo izvrši rotacija desnog, odnosno lijevog čvora djeteta oko roditelja u desnom podstablu čvora, a zatim se taj čvor rotira sa lijevim, odnosno desnim djetetom. Ove funkcije koriste se kod dodavanja ili brisanja čvora u slučaju kada je $FR(P,Q) = (2, -1)$, ili $FR(P,Q) = (-2, 1)$, gdje je P čvor roditelj, a Q desno, odnosno lijevo dijete.

Za dobivanje visine nekog čvora koristi se funkcija *height*. Ako je čvor prazan, funkcija vraća vrijednost -1. U protivnom, vraća se vrijednost visine čvora. Ispis elemenata nije dio klase AVL, obzirom da ne ovisi o implementaciji samog stabla. Stoga je metoda *printTree* odvojena od funkcionalnosti stabla, i nalazi se zajedno s ostalim metodama sučelja.