Fakultet elektrotehnike i računarstva

Zavod za primjenjeno računarstvo

**Napredni algoritmi i strukture podataka**

1. laboratorijska vježba

Vinko Kodžoman 0036470625

Zagreb, 4.11.2015

**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. Treba implementirati i brisanje čvorova u stablu za dodatnih 6 bodova.

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

**Rješenje zadatka**

**Teorijski uvod**

Organizacija podataka i algoritmi koji se vrte nad tim podacima su jezgre većine programa danas. Stabla su pogodna za organizaciju podataka jer unatoč većem broju memorije (npr. nego polja), omogućuju brže operacije dodavanja, brisanja i pretraživanja elemenata()). Česte operacije mogu narušiti strukturu stabla i oduzeti njegova pogodna svojstva. Samo balansirajuća stabla sama održavaju svoju strukturu stabla i ne dopuštaju veliko narušavanje strukture. Danas dva najpopularnija samo balansirajuća stabla su AVL i crveno crno stablo. AVL stablo osigurava manju visinu (maksimalna udaljenost od čvora do lista stabla) od crveno crnih stabla, samim time je pogodnije za pretraživanje podataka od crveno crnih stabla. Prednost crveno crnih stabla je u operacijama dodavanja i brisanja čvorova unutar stabla koje su jednostavnije od onih u AVL stablima. Iz gornjeg sljedi da se crveno crna stabla koriste nad skupom podataka sa učestalim dodavanjem i brisanjem elemanata skupa.

Kako bi osiguralni sva gore navedena svojstva moramo se pridržavati nekih pravila. Stabla će biti binarna (svaki čvor ima dvije djece – lijevo i desno dijete), lijevo podstablo će sadržavati sve brojeve manje ili jednake od elementa u promatranom čvoru, a desno podstablo sve brojeve veće. Ovime smo ostvarili BST (binary search tree). Pridržavajući se pravila za crveno crna stabla dobivamo samo balansirajuće BST.

**Implementacija**

Program je napisan u programskom jeziku C++ u Visual Studiu 2015. Niti jedna vanjska biblioteka nije korištena, osim onih ugrađenih u C++11 standard (std, algorithm, ...), compiler gcc.

1. **Struktura čvora**

BST pretvaramo u crveno crno stablo dodavanjem još jednog atributa u strukturu stabla – boja (enumeracija koja prima vrijednosti RED, BLACK). Korištenjem posebnih pravila za rad na bojama u čvorovima osiguravamo balansiranost stabla.

1. **Organizacija čvorova**

Organizacijom čvorova na poseban naćin dobivamo pogodna svojstva. Svaki čvor ima dvoje djece (binarno stablo), u lijevo podstablo se nalaze svi elementi manji ili jednaki, a u desno veći od elementa u promatranom čvoru. Jedno od svojstva je sortirani ispis stabla inorder prolazom za uzlazno sortirane elemente ili za naći najmanji element u skup moramo otići u najljeviji čvor (najdesniji za najveći element).

1. **Pravila crveno crnih stabla**

* Svaki čvor je crven ili crn
* Korijen je crn
* Oba potomaka crvenog stabla su crna
* Svaki list je crn – prazan (NIL) također
* Svaka staza od bilo kojeg čvora do lista koji je njegov potomak mora proći kroz jednak broj crnih čvorova

1. **Inorder ispis i rotacij**

Održavanjem stabla kao BST (lijevo podstablo sadrži sve elemente <=, a desno > od elementa u trenutno promatranom čvoru) omogućuje sortirani ispis svih elemenata inorder obilaskom stabla.

inorderIspis(Node cvor) {

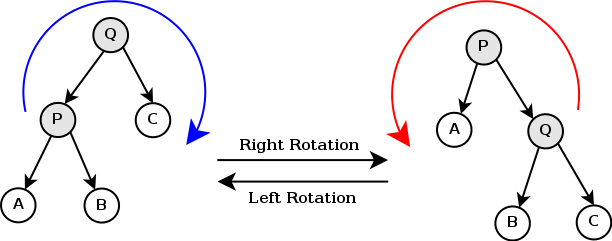
inorderIspis(cvor -> lijevoDijete);

ispis(cvor -> element);

inorderIspis(cvor -> desnoDijete);

}

Važno je da gore spomenutu svojstvo ostane sačuvano nakon izvođenja operacija nad stablom pri uravnotežavanju stabla. Rotacija je temljna operacija koja se izvodi nad stablom, omogućuje promijenu izgleda stabla bez mijenjanja ispisa stabla u prolasku kroz stablo (preorder, inorder i postorder) [Slika 1].



Slika 1 Primjeri lijeve (left) i desne (right) rotacije u stablu

Nakon rotacije oba stabla imaju isti inorder ispis (APBQC) [Slika 1], dok im je struktura (izgled) promijenjen. Primjenom rotacija nakon dodavanja i brisanja čvorova u stablu možemo popraviti sve čvorove koji naruše pravila crveno crnih stabla [3.].

1. **Dodavanje i brisanje čvorova u crveno crnom stablu**

Za dodati novi element u crveno crno stablo primjenjuje se BST pravila (lijevo <=, denso >), na kraju spusta dodaje se novi čvor. Nakon dodavanja čvora algoritam se vraća od novog čvora do korjena kako bi ispravio sve nepravilnosti koje su nastale dodavanjem novog čvora. Nepravilnosti su ispravljene rotacijama i eventualnim promijenama boja pojedeinih čvorova. Ekvivalentno vrijedi i za brisanje, stablo se ispravlja nakon uspješno dovršene operacije brisanja čvora.

**Zaključak**

Da bismo riješili problem moramo izabrati pravi alat, danas postoji pretek struktura podataka i algoritma i kao programeri moramo znati koji alat upotrijabiti. Crveno crna stabla sama održavaju ravnotežu stabla (do dva puta visine savršeno uravnoteženog stabla [1]) i time omogućavaju brza operacije pretrage, dodavanja i brisanj elemenata u stablo. Prednost crveno crnih stabala nad AVL stablima (vrsta samo balansirajućih stabala) je manja složenost operacija dodavanja i brisanje, ali zato veća složenost pretraživanja i dohvata zbog veće visine crveno crnog stabla. Zbog gore navedenih svojstava, crveno crna stabla se koriste za pohranu podataka sa učestalim dodavanjem i brisanjem novih elemenata.

**Literatura**

|  |  |
| --- | --- |
| **Vrsta** | **Format** |
| Knjiga | Domagoj Kusalić: Napredno programiranje i algoritmi u C-u i C++-u 5. nepromijenjeno izdanje, Element, Zagreb, 2014.  Julijan Šribar, Boris Motik :Demistificirani c++ 4. dopunjeno izdanje, Element, Zagreb, 2014. |
|  |  |