

Crveno crna stabla

Nikola Nešić

October 7, 2024

1 Opis teme

Crveno crna stabla su stabla za koja važi sledeće:

1. Svaki čvor je ili crven ili crn.
2. Koren je crn
3. Svi listovi su crni i ne sadrže vrednost(označavamo ih sa null)
4. Svi crveni čvorovi imaju tačno dva crvena deteta
5. Sve putanje od nekog čvora do listova u njegovom poddrvetu sadrže isti broj crnih čvorova.

Bavićemo se implementacijom crveno crnih stabala i osnovnim operacijama nad njima to jest pretragom,ubacivanjem vrednosti u stablo i vadjenjem vrednosti iz stabla.

2 Implementacija

2.1 Pretraga

Imamo **traženu vrednost** za koju nas zanima da li se nalazi u stablu, i **trenutni cvor** koji je koren stabla u kome pretražujemo. Spuštamo se niz stablo od korena do lista na dole.

- i) Ako je trenutni čvor null onda se tražena vrednost ne nalazi u stablu.
- ii) Ako je vrednost čvora jednaka traženoj vrednosti ,vrednost se zaista nalazi u stablu.
- iii-r) Ako je tražena vrednost veća od vrednosti čvora vrednost se nalazi u stablu akko se nalazi u desnom podstablu čvora .
- iv-r) Ako je tražena vrednost manja od vrednosti čvora onda se nalazi u stablu akko se nalazi u levom podstablu.

2.2 Ubacivanje vrednosti

Ubacivanje se sastoji iz dva koraka:)

- ubacivanja čvora
- korekcija stabla

2.2.1 Ubacivanje čvora

Imamo **vrednost koju ubacujemo** u stablo i **čvor koren stabla** u koje ubacujemo.

- i) Ako je vrednost čvora jednaka traženoj vrednosti ,vrednost se već nalazi u stablu i ne treba je ubacivati.
- ii) Ako je čvor null onda se tražena vrednost ne nalazi u stablu,i treba je ubaciti umesto trenutnog čvora kao crveni čvor sa dva crna null deteta.

iii-r) Ako je vrednost koju ubacujemo veća od vrednosti čvora vrednost treba ubaciti u desno podstablo.

iv-r) Ako je vrednost koju ubacujemo manja od vrednosti čvora onda se vrednost treba ubaciti u levo podstablo.

Primetimo da samo u slučaju ii) može doći do narušavanja pravila, i to samo pravila 2 ili 4. Može biti potrebno izvršiti korekciju stabla.

2.2.2 Korekcija stabla

Korekciju vršimo ako je narušen uslov 2 (ako nema roditelja) ili 4 (ako mu je otac crven). Za izvršavanje korekcije neophodno je da znamo roditelja čvora (**P**), dedu čvora (**G**) i ujaka čvora (**U**):

- i) Ako je prekršen uslov 2 neophodno je samo obojiti koren u crno.
- ii-r) Ako je **P** crven i **U** crven, možemo obojiti **P** i **U** u crnu, a **G** u crveno, sada je **G** potencijalno prekršio uslove pa nad njim potencijalno treba izvršiti rekursivno korekciju.
- iii))Ako je **U** crn, a on je dete bliže unutrašnjosti, neophodno je popeti ga odgovarajućom rotacijom nad **P** i svesti na slučaj iv).
- iv) Ako je **U** crn, a on je dete bliže spoljašnjosti, neophodno je popeti **P** odgovarajućom rotacijom, prebojiti **G** u crveno a **P** u crno.

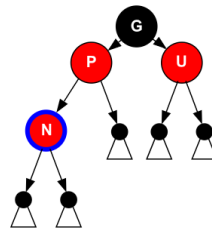


Figure 1: ii-r) pre korekcije.

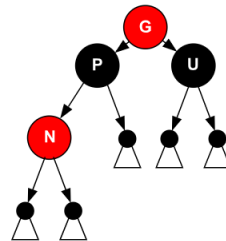


Figure 2: ii-r) posle korekcije.

2.3 Brisanje vrednosti

- brisanje čvora
- korekcija stabla

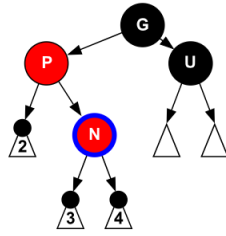


Figure 3: iii) pre korekcije.

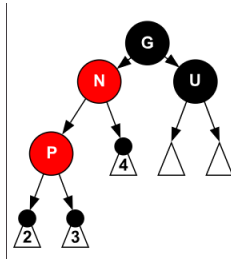


Figure 4: iii) posle rotacije.

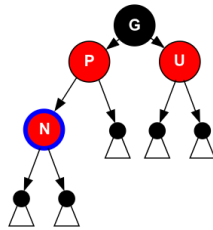


Figure 5: iv) Pre.

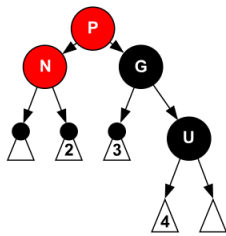


Figure 6: iv) posle rotacije.

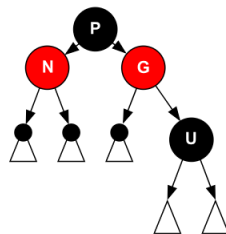


Figure 7: iv) posle bojenja.

2.3.1 Brisanje čvora

Imamo vrednost za brisanje i koreni čvor stabla.

- i) Ako je čvor null vrednost se svakako ne nalazi u stablu.
- ii) Ako je vrednost čvora jednaka vrednosti za brisanje i ako je čvor list, treba ga zameniti null čvorom. Ako je čvor bio crn treba null čvor označiti duplo crnim i izvršiti korekciju.
- iii) Ako je vrednost čvora jednaka vrednosti za brisanje i ima jedno dete (to dete mora biti crveno). Menjamo njega njegovim detetom i bojimo ga u crno.
- iv) Ako je vrednost čvora jednaka vrednosti za brisanje i ako ima oba deteta, i ako čvor nije list, treba da trenutni čvor zameni vrednost sa vrednošću njegovog najlevljjeg lista desnog stabla. Zatim iz desnog podstabla treba obrisati vrednost za brisanje.
- v-r) Ako je vrednost za brisanje veća od vrednosti čvora onda treba obrisati vrednost za brisanje iz desnog podstabla.
- vi-r) Ako je vrednost za brisanje manja od vrednosti čvora onda treba obrisati vrednost za brisanje iz levog podstabla.

2.3.2 Korekcija stabla

Za korekciju neophodno je znati njegovog (duplo crnog čvora) oca (**P**), brata (**S**) i sestriće (blizeg (**C**)) i daljeg (**D**))

- i) Ako je on koren samo ga farbamo u crno.
- ii-r) Ako su **P**, **S**, **C** i **D** crni, Možemo ga obojiti u crno, **P** u duplo crno, **S** u crveno. Zatim rekursivno izvršiti korekciju za **P**.
- iii) **P** je crno, **S** je crveno a **C** i **D** su crni: Neophodno je podići **S** odgovarajućom rotacijom. Zatim **S** obojiti u crno a **P** u crveno. Zatim se svodi na jedan od slučajeva iv, v ili vi.
- iv) **P** je crven, **S** je crn a **C** i **D** crni: **P** farbamo u crno, njega farbamo u crno, a **S** u crveno.
- v) **P** je proizvoljne boje, **S** je crne, **D** je crn, **C** je crven: neophodno je popeti **C** odgovarajućom rotacijom, zatim obojiti **C** u crno a **S** u crveno. Zatim se direktno svodi na slučaj vi.
- vi) **P** je proizvoljne boje, **S** je crn, **D** je crven. Treba popeti **S** odgovarajućom rotacijom, zatim obojiti **S** u boju od **P**, **P** obojiti u crno i **D** u crno.

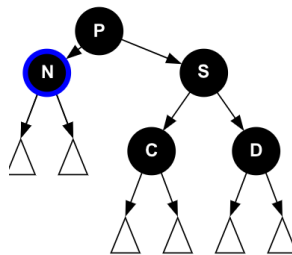


Figure 8: ii-r) pre korekcije.

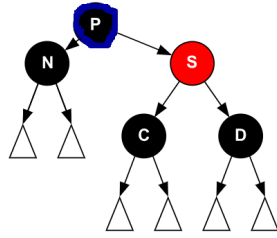


Figure 9: ii-r) posle korekcije.

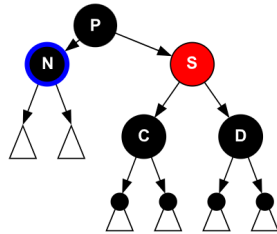


Figure 10: iii) pre korekcije.

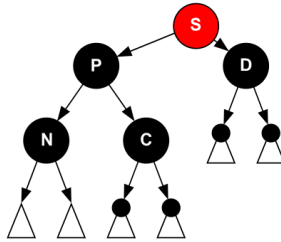


Figure 11: iii) posle rotacije.

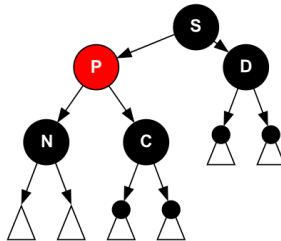


Figure 12: iii) posle bojenja.

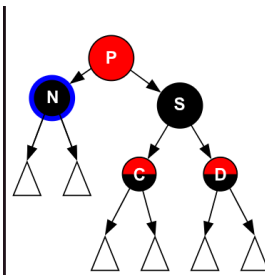


Figure 13: iii) posle korekcije.

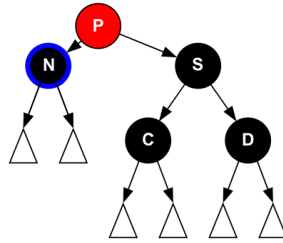


Figure 14: iv) pre korekcije.

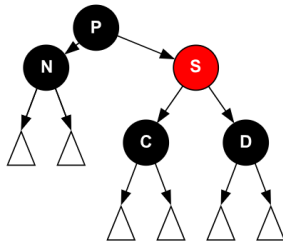


Figure 15: iv) posle korekcije.

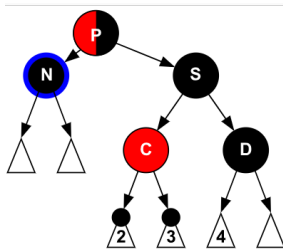


Figure 16: v) pre korekcije.

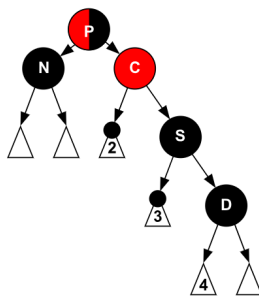


Figure 17: v) posle rotacije.

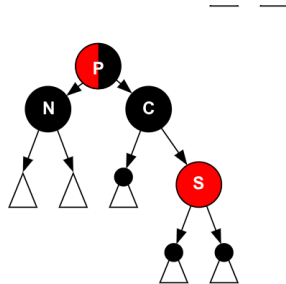


Figure 18: v) posle bojenja.

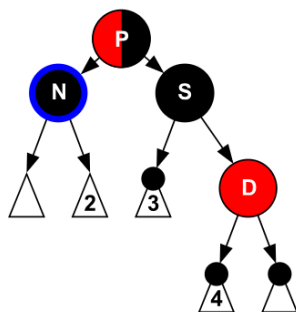


Figure 19: vi) pre korekcije.

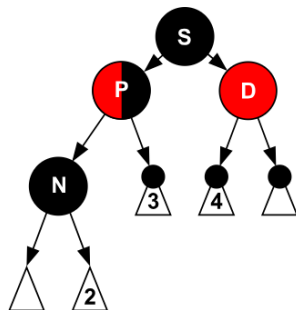


Figure 20: vi) posle rotacije.

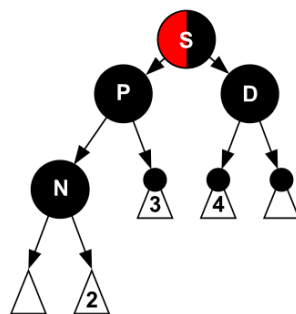


Figure 21: vi) posle bojenja.

2.4 Analiza složenosti

Sve operacije nad stablom su složenosti $O(h)$ stabla jer se spuštamo od korena do lista i vraćamo nazad do korena kroz rekurziju. Ako dokažemo da je $h = O(\log(n))$ onda je složenost svih operacija $O(\log(n))$.

Neka je:

1. $h(v)$ visina poddrveća čiji je koren čvor v , tj. broj čvorova od čvora v do najdaljeg lista (ne računajući čvor v).
2. $h_b(v)$ – crna visina poddrveća čiji je koren čvor v , tj. broj crnih čvorova od čvora v do proizvoljnog lista (ne računajući čvor v ako je on crn).

Lema:

$n \geq 2^{h_b(v)} - 1$. Odnosno da je minimalan broj čvorova stabla crne visine h_b , $n_{min} = 2^{h_b(v)} - 1$. Crveno crno stablo sa minimalnim brojem čvorova je stablo sa isključivo crnim čvorovima, zbog uslova 1-5 to je potpuno balansirano binarno stablo visine h_b . Pa je broj čvorova takvog stabla $n_{min} = 2^{h_b} - 1$.

Dakle važi $n \geq 2^{h_b(v)} - 1$. Odnosno $h_b(v) \leq \log_2(n + 1)$.

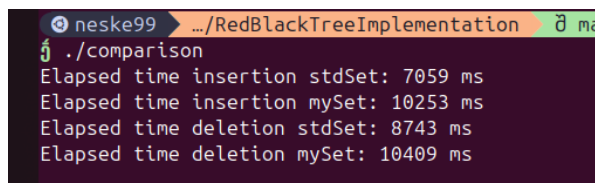
Dokaz:

Pošto je bar pola čvorova na svakoj putanji od korena v do listova crno, važi da je $h_b(v) \geq h(v)/2$ sledi $h(v)/2 \leq \log_2(n + 1)$ odnosno $h(v) \leq 2\log_2(n + 1)$. Što je i trebalo pokazati.

3 Analiza implementacije i poredjenje performansi sa C++ standardnom bibliotekom

Naše unosenje je oko 45 procenata sporije od standardnog, a brisanje oko 20 procenata. Zašto?

Kod nas je vrednost koja se čuva u čvoru predstavljena Optional-om tipa koji se čuva, a null čvorove crnim čvorovima sa praznom Optional vrednošću. Dakle za sve null listove alociramo dodatnu memoriju. U najgorem slučaju potpunom balansiranom stablu ovo je čak duplo više alokacija memorije (+1).



```
neske99 ~/RedBlackTreeImplementation 0 ma
$ ./comparison
Elapsed time insertion stdSet: 7059 ms
Elapsed time insertion mySet: 10253 ms
Elapsed time deletion stdSet: 8743 ms
Elapsed time deletion mySet: 10409 ms
```

Figure 22: Primer poredjenja rezultatata. Uzimamo permutaciju brojeva od 0 do 5 miliona i ubacujemo ih redom te permutacije, pa ih brišemo redom iz druge permutacije iz odgovarajućih struktura podataka i poredimo vreme izvršavanja za te korake.

References

[Wiki](#)
[Youtube](#)
[Matf](#)