

STABLA OPSTEG PRETRZIVANJA

Stablo binarnog pretraživanja nije pogodno za organizaciju indeksnih struktura, jer jedan pristup disku služi samo za binarno odlučivanje na koje od dva podstabla otići za naredni pristup. Da bi se povećala efikasnost, poželjno je da blok, koji se učitava jednim pristupom, sadrži više informacija koje omogućavaju efikasnije odlučivanje o proizvodu pretraživanja ne samo između dva nego između više podstabala.

STABLA M-ARNOG PRETRAZIVANJA:

Stepen svakog čvora je manji ili jednak sa m , i zadovoljene su sledeće osobine

- Čvor se sastoji od pokazivaca na podstabla ($p_i, 0 \leq i \leq k$) i ključeva ($K_i, 1 \leq i \leq k$) koji su organizovani tako da svaki ključ razdvaja dva pokazivaca gde je $1 \leq k \leq m$
- Svi ključevi u čvoru su rastuće uređeni
- Svi ključevi u podstablu na koje pokazuje p_{i-1} levo od ključa K_i su manji od tog ključa, a oni na koje pokazuje p_i svi su veći
- Podstablo na koje pokazuje pokazivac p_i je takođe stablo m -arnog pretraživanja.

Obilazak: je najčešće po inorderu jer se tako dobijaju ključevi u rastućem poretku. **Umetanje:** Novi ključ se uvek umeće u list, postajući logički poredak. Umetanju uvek prethodi neuspelo pretraživanje. Ako list nije potpuno popunjen svi veći ključevi se pomeraju i novi ključ se umeće na mesto gde je bio prvi veći ključ. Pokazivaci se ne diraju jer se radi o listu. Ako je list popunjen onda se stvara novi list na koji pokazuje pokazivac između prvog manjeg i prvog većeg ključa od zadatog. Ovaj popunjeni list prestaje da bude list i postaje čvor. Umetanje garantuje da se novi list sa sledbenicima stvara dok se prethodni čvor - list ne popuni. Stablo se popunjava od gore na dole i zato se zove **top-down multiway tree**. Ovde je dobra osobina što je veliki broj ključeva na visim nivoima, pa im se brže pristupa. **Brisanje** ako je ključ u listu list se sažima. Ako je ključ jedini u listu onda se taj list dealocira. Ako se uklanja iz čvora, a ima levo i desno podstablo prazno onda se ključ zajedno sa levim i desnim pokazivacem ukloni iz čvora. Ako ima oba podstabla nalazi se sledbenik, kao najmanji u desnom podstablu, pa se obrisani ključ menja svojim sledbenikom. Postupak se nastavlja dok se ne dođe do lista. **Osobine** : Iskoriscenost je oko 50%. Maksimalna dužina pretraživanja odedjena je visinom stabla, a na svakom nivou se ispituje najviše jedan čvor.

$h_{\min} = \lceil \log_m(n+1) \rceil - 1$. Složenost algoritma pretraživanja je $O(\log_m n)$

Implementacija: U slučaju ključeva različitih dužina. Na početku svakog čvora se čuvaju pozicije početka svakog ključa u čvoru. Sada je teže umetanje i brisanje. U slučaju da su prvi znakovi više ključeva isti onda se prvi zajednički znakovi mogu pamti samo jednom, a za sve ostale pamte samo znakovi koji ih razlikuju. Ovo se naziva **ceona kompresija**.

B - STABLA : reda m se definise kao m -arno stablo sa sledecim svojstvima: •Koren stabla ima najmanje dva podstabla; •svi cvorovi osim korena i listova imaju najmanje $\lceil m/2 \rceil$ podstabala; •listovi imaju najmanje $\lceil m/2 \rceil - 1$ kljuceva; •svi listovi suna istom nivou; Ovim se garantuje da je svaki cvor osim korena bar polupopunjen pa je iskoriscenost prostora bolja. Kako su svi listovi na istom nivou to je obezbedjena balansiranost stabla. **Pretrazivanje**: pocevsi od korena ucitava se sadrzaj jednog cvora, pa se u opreativnoj memoriji njegovi kljucevi uporedjuju sa zadatom vrednoscu K . Ukoliko se nadje ist operacija ja uspesna, a ako ne nadju se kljucevi K_{i-1} i K_i koji zadovoljavaju uslove $K_{i-1} < K < K_i$ pa se pretrazivanje na isti kljuc nastavlja u cvoru na koji pokazuje pokazivac p_i . Ako je $K < K_1$ cita se cvor ukazan sa p_0 , a ako je $K > K_j$ onda se cita cvor na koji ukazuje p_j i u njemu nastavlja pretrazivanje. Pretrazivanje je neuspesno ako se dodje do praznog podstabla. $h_{\min} \leq \log_{\lceil m/2 \rceil} (n+1)/2$ pa je i slozenost pretrazivanja $O(\log_{\lceil m/2 \rceil} n)$. **Umetanje**: Prvo se pokusava umetnuti u list. Ako list nije pun, novi kljuc se umetne po sortiranom poretku, pa se veci kljucevi pomere za jedno mesto navise i sadrzaj list ponovo upise na disk. Pokazivaci se ne diraju jer su prazni. Ako je list pun ($m-1$ kljuceva) onda dolazi do njegovog preloma. Alocira se novi cvor sa adresom $P1$. Formira se rastuci niz od m kljuceva koji su ranije bili u starom listu P . Taj niz se prelama oko srednjeg kljuca $K_{\lceil m/2 \rceil}$. Prelom se vrši tako sto se kljucevi $K_1, \dots, K_{\lceil m/2 \rceil - 1}$ zadržavaju u starom listu P , a kljucevi $K_{\lceil m/2 \rceil}, \dots, K_m$ idu u novi list $P1$. $K_{\lceil m/2 \rceil}$ se salje u roditeljki cvor. Ako roditeljski cvor nije pun ovaj kljuc se smesta u njega na mesto koje mu odgovara po sortiranom poretku. Ako je roditeljski cvor pun onda se i kod njega vrši prelim i tako do korena. U najgorem slucaju kad se na svakom od $h+1$ nivou desi prelom (koji zahteva dve operacije upisa na disk) pa je, ukljucujuci i prethodno citanje ukupno $3(h+1)+1$ pristup disku. Maksimalni broj preloma po jednom kljucu je $(n-2)/(1+(n-1)(\lceil m/2 \rceil - 1))$ sto se za veliko n svodi na $1/(\lceil m/2 \rceil - 1)$. **Brisanje**: Ako je kljuc u listu, i posle uklanjanja trazenog kljuca u listu ostane dovoljan broj kljuceva (bar $\lceil m/2 \rceil - 1$), operacija se završava nakon pomeranja kljuceva vecih od obrisanog za jedno mesto u levo. Medjutim, ako je broj preostalih kljuceva manji od minimuma, onda se proverava da li desni brat R cvora P ima bar jedan kljuc vise od minimuma. Ako ima, onda kljuc iz cvora roditelja koji razdvaja ova dva brata prelazi u P i popunjava ga do minimuma, a najmanji kljuc iz R prelazi na njegovo mesto u roditeljskom cvoru. Ako desni ima minimalno kljuceva moze se analogno pokusati sa levim bratom. Ukoliko se ne moze preuzeti kljuc od brata vrši se operacija spajanja. Kod ove operacije $\lceil m/2 \rceil - 2$ kljuceva preostalih posle brisanja u listu P , razdvojni kljuc iz roditeljskog cvora i $\lceil m/2 \rceil - 1$ kljuceva iz brata R se smestaju u cvor P . Sada se dealocira R i iz roditeljskog cvora ukloni pokazivac na R . Ako se pri spajanju desi da i roditeljski cvor ostane sa nedovoljnim brojem kljuceva spajanje se nastavlja ka vrhu stabla. Ako se kljuc koji se brise ne nalazi u listu, vec u cvoru, taj kljuc menja svoje mesto sa kljucem sledbenikom koji je u listu, i tada se brisanje svodi na brisanje iz lista. U najgorem slucaju kada se spajanje propagira do korena broj pristupa disku je srazmeran trostrukoj visini stabla. Prosečna popunjenost je oko 70%

B* - STABLA: B stablo reda m ima sledece osobine: •Svaki cvor osim korena ima najviše m podstabala; •svaki cvor osim korena i listova ima najmanje $(m-1)/3$ podstabala; •koren ima najmanje 2, a najviše $2\lfloor(2m-2)/3\rfloor+1$ podstabala; •svi listovi su na istom nivou; •cvor, ako nije list, sa k podstabala ima $k-1$ kljuceva. B*-stablo forsira popunjenost oko $2/3$. Umesto prelamanja kao kod B stabla ovde se pokusava prelivanje kljuceva i preraspodela sa susedima. Tada se prvo proverava broj kljuceva u desno bratu punog cvora. Neka on ima $j < m-1$ kljuceva sto znaci da nije pun. Tada se svi kljucevi iz oba brata i razdvojni kljuc posmatraju kao jedan niz, pa se zatim $\lfloor(m+j+1)/2\rfloor$ manjih kljuceva ostavi u cvoru koji je bio pun, srednji ide u roditeljski, a preostali u desnog brata. Ako nema desnog brata ili je pun pokusava se sa levim. Ako je i on pun prelom je neizbezan. Sada se prelamaju dva puna cvora od jednom. Kljucevi iz oba cvora, razdvojni iz roditeljskog i onaj koji se umece se rasporeduju na sledeci nacin: U prvi cvor ide $\lfloor(2m-2)/3\rfloor$, u drugi $\lfloor(2m-1)/3\rfloor$, a u treci $\lfloor 2m/3\rfloor$, dok u roditeljski cvor idu dva razdvojna kljuc. Ako dodje do prepunjenja roditeljskog cvora postupak se ponavlja do korena. Ako se prepuni koren od njega nastaju.

B+ - STABLA: Uvode se modifikacije za B stablo i tako nastaje B+ stablo. Cvorovi grananja i listovi se razlikuju po svojoj organizaciji. Cvorovi: •sadrze kljuceve i pokazivace na podstabla kao i u B stablu (ali ne i pokazivace na zapise koji odgovaraju ovim kljucevima), pri cemu je najveći broj podstabala m , a najmanji $\lceil m/2 \rceil$, samo koren moze da ima minimalno dva podstabala; •kljucevi u okviru cvora su uredjeni $K_i < K_j$ za $i < j$; •za svaki kljuc u podstablu ukazanom pokazivacem p_i vazi da je veci od K_{i-1} , a manji ili jednak od K_i . Listovi: •sadrze najmanje $\lfloor m/2 \rfloor$ kljuceva u uredjenom poretku; •ne sadrze pokazivace na podstabla vec pokazivace na odgovarajuce zapise; •svi listovi su na istoj dubini; •svi listovi su uredjeni u ulancanu listu poodgovarajucem logickom poretku. Neki kljucevi mogu da se javu dva puta u stablu. Tako se svaki kljuc K_i koji se nalazi u cvoru grananja ponavlja u listu i to kao najdesniji, najveći kljuc u podstablu ukazanom pokazivacem levo od njega p_{i-1} . **Pretrazivanje:** Za razliku od B stabla gde se uspesno pretrazivanje moze završiti na odredjenom nivou, ovde se i uspesno i neuspesno pretrazivanje završava na poslednjem nivou, u listu, jer samo tamo postoji adresa za pristup zapisu. Maksimalna visina B+ reda m je $h_{\min} = \log_{\lceil m/2 \rceil} (n+1)/2$. **Umetaanje:** Novi kljuc se uvek umece u list. Ako se list prepuni dolazi do preloma, stvara se novi list, $\lceil m/2 \rceil$ kljuceva se zadržava u starom listu, najveći od njih se replicira kao razdvojni u roditeljskom cvoru i dodaje mu se pokazivac na novi list, a ostali kljucevi idu u novi list. Ako se na ovaj nacin prepuni roditeljski cvor prelom se radi kao kod B stabla. **Brisanje:** Kljuc se uvek uklanja iz lista, a ako je najdesniji onda se uklanja i iz cvora grananja, pri cemu ga zamenjuje njegov levi sused (prethodnik). Ukoliko broj kljuceva u cvoru padne ispod minimuma pokusava se preuzimanje iz susednog cvora. Da bi se brisanje uprostilo dovoljno je obrisati kluc iz lista a ostaviti njegovu repliku.

STBLA DIGITALNO PRETRAZIVANJA:

DIGITALNO STABLO: Struktura se zasniva na sumi opstih stabala stepena jednakog broja mogućih različitih znakova u ključu (npr. cifre 0..9), gde su svi ključevi sa istim prvim znakom organizovani u isto stablo. U svakom stablu ključevi se sastoje od znakova koji se nalaze na putevima od korena do listova. Postoje ključevi mogu biti različite dužine postoji poseban znak EOK koji ukazuje na kraj ključa i postoji samo u listovima. **Pretraživanje:** Pri pretraživanju tretira se da ključ kao poslednji znak sadrži EOK. Pretraživanje se vrši od nultog nivoa tako da se traži cvor koji sadrži prvi znak ključa. Ako se nađe saglasnost prelazi se na sledeći nivo koji se pretražuje na drugi znak ključa itd. Uspešno pretraživanje se završava u listu gde se nalazi adresa odgovarajućeg zapisa. Prvi put kad se na određenoj nivou ne nađe cvor sa znakom koji se u ključu nalazi na odgovarajućoj poziciji, pretraživanje se proglašava neuspešnim. **Umetanje:** Umetanje ključa K započinje pretragom. Kada se dođe do nivoa $i-1$ na kojem ne postoji cvor sa znakom $K[i]$, takav cvor se stvara i uvezuje u listu braca na istom nivou, a zatim se ispod njega dodaje po jedan cvor na odgovarajućem nivou za sve preostale znakove ključa. Postoji i modifikacija koja smanjuje broj cvorova. U slučaju kad od određenog cvora postoji put do samo jednog lista, onda se svi cvorovi ispod njega mogu izbaciti, a u tom cvoru odmah zapamtiti adresu na odgovarajući zapis. Ova organizacija je efikasna a ako cvor ima relativno malo sinova inace je rešenje->

TREE - STABLO: Ako se odlučivanje u jednom cvoru umesto na osnovu jednog bita vrši u odnosu na jedan znak ključa dolazi se do TREE stabla. Ako se ključevi grade od skupa od m znakova onda se cvor realizuje kao niz od $m+1$ pokazivaca. Svaki od m pokazivaca poziciono odgovara jednom znaku, a $(m+1)$ -vi odgovara znaku za kraj ključa EOK. Pokazivac p_i u korenu ukazuje na podstablo koje registruje sve ključeve čiji je prvi znak i ($i\dots$), zatim pokazivac p_j u korenu tog podstabla ukazuje na podstablo sa ključevima koji počinju sa ij ($ij\dots$). Grananje na nivou $k-1$ se vrši na osnovu znaka ključa na poziciji k . **Pretraživanje:** se vrši tako što se na svakom narednom nivou konsultuje sledeći znak ključa i ispita odgovarajući pokazivac. Pretraživanje je neuspešno čim se na putu od korena ka listu nađe na prazan pokazivac, a nisu ispitani svi znakovi ključa. **Umetanje:** započinje neuspešnim pretraživanjem. Za umetnuti ključ treba da se uspostavi lanac pokazivaca kroz cvorove koji je dugacak koliko i ključ. Ako u stablu već postoji ključ sa istim početnim znakovima, deo tog lanca postoji i treba ga samo nastaviti aktiviranjem novih pokazivaca na preostale znakove. **Brisanje:** Prvo se locira cvor u kojem se nalazi adresa zapisa sa datim ključem kao poslednji pokazivac u cvoru, pa se ova adresa zameni praznim pokazivcem. Zatim se proveriti da li imaju aktivnih pokazivaca u cvoru. Ako nema dati cvor se dealocira. Potom se ista proveriti radi i na roditelju. Postupak se završava ukoliko se nađe prvi predak koji ima bar jedan

aktivan pokazivac. Ovo je efikasna struktura kada je skup kljuceva gust, jer je vecina pokazivaca u cvoru iskoriscena. Pogodno je za kljuceve promenjive duzine.