

Napredni algoritmi i strukture podataka

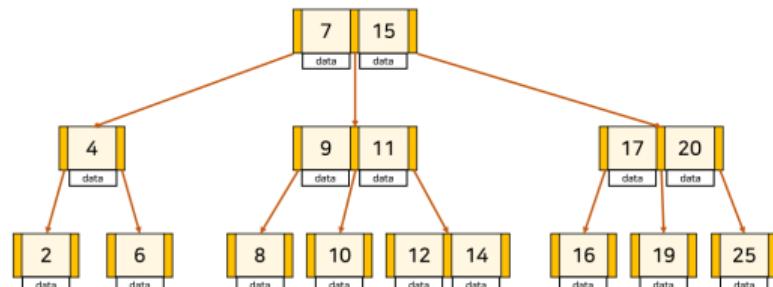
B stablo



Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka

B stablo

- ▶ B stablo je stablo pretrage koje može da sadrži više ključeva unutar jednog čvora
- ▶ Ključevi u čvoru su sortirani u rastućem redosledu
- ▶ Između svaka dva ključa nalazi se podstablo sa ključevima koji su veći od levog i manji od desnog ključa
- ▶ Pre prvog/posle posljednjeg ključa nalaze se podstabla sa manjim/većim ključevima



Svojstva

- ▶ B stablo nasleđuje svojstvo stabala pretrage da se za svaki čvor u levom podstabalu nalaze samo čvorovi sa manjim elementima, a u desnom podstabalu samo čvorovi sa većim elementima
- ▶ Pored toga, za B stablo reda **m** važi da:
 - ▶ Svaki čvor sadrži do m dece i do $m-1$ ključeva
 - ▶ Svaki čvor (sem korena) sadrži barem $b = \lceil m/2 \rceil$ dece i barem $b - 1$ ključeva
 - ▶ Svaki list je iste dubine
- ▶ Ovakva svojstva osiguravaju da je svaki čvor (sem korena) barem polovično popunjen
- ▶ Maksimalna dubina stabla sa N elemenata je $1 + \log_b((N + 1)/2)$ -> vreme pretrage, dodavanja i brisanja je $O(\log N)$

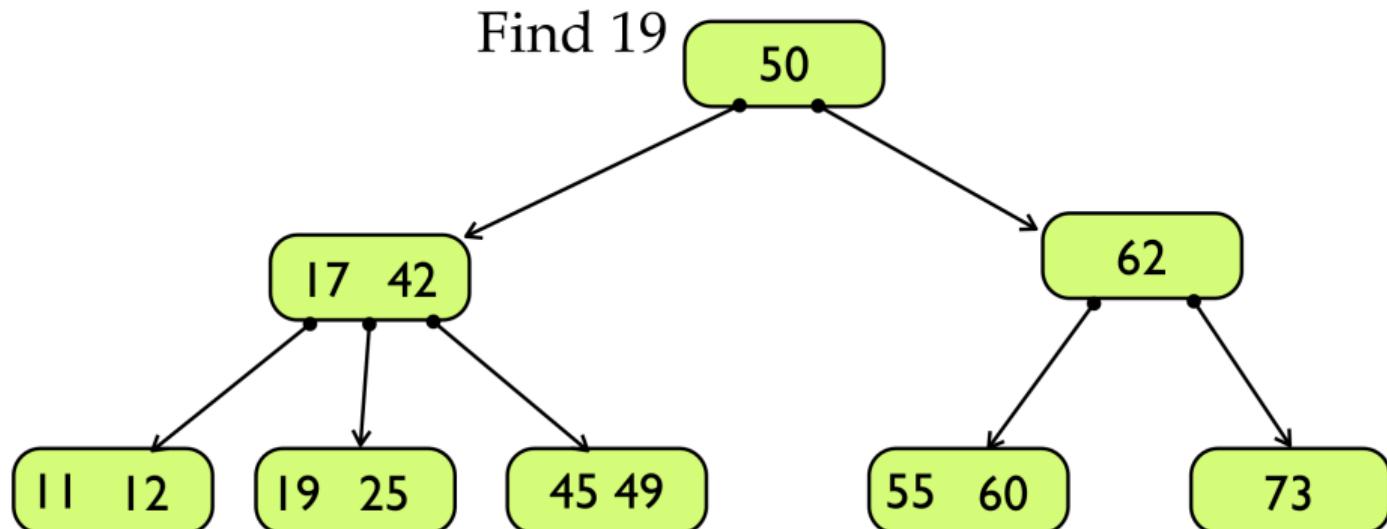
B stablo vs binarno stablo pretrage

- ▶ Vreme izvršavanja svih operacija zavisi od dubine stabla
- ▶ U opštem slučaju, faktor grananja je kod B stabla veći nego kod binarnog stabla pretrage, što znači da će dubina B stabla biti manja
- ▶ Koristimo ih kao efikasnu strukturu u memoriji kada je broj podataka jako veliki, ili još češće za indeksiranje fajlova na disku

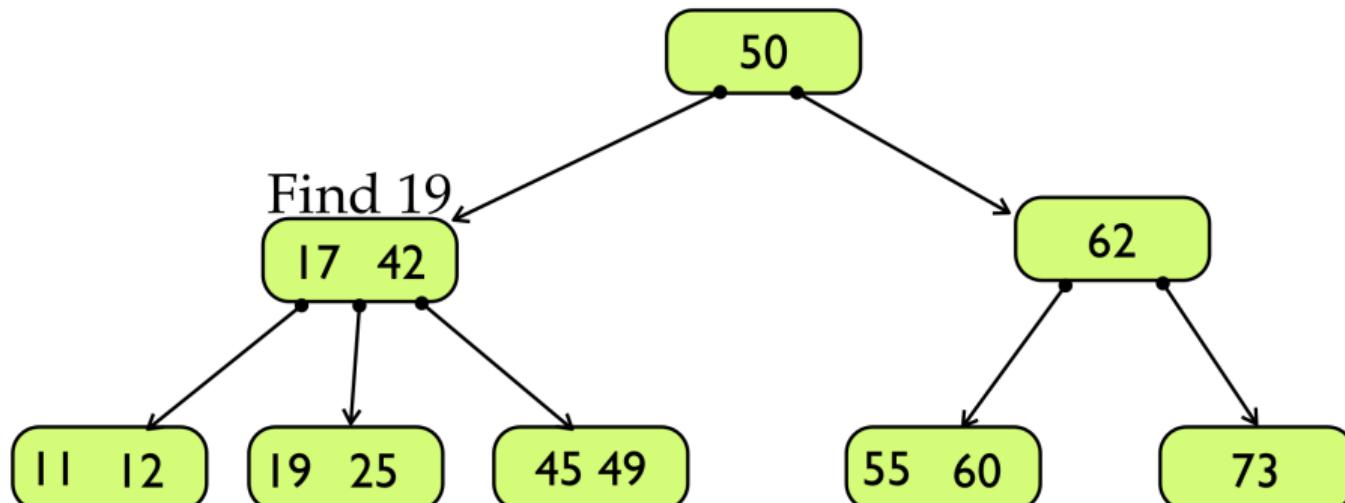
Pretraga

- ▶ Pretraga karakteristična za bilo koje stablo pretrage (manji elementi levo od trenutnog, veći su desno)
- ▶ Tražeći ključ key , krenemo od korena i proveravamo svaki ključ k u čvoru:
 - ▶ $key == k$ — pronašli smo traženi ključ
 - ▶ $key < k$ — pređi na čvor koji je $index(k)$ -to dete trenutnog čvora
 - ▶ $key > k$ AND no next k — pređi na čvor koji je $index(k)+1$ -to dete trenutnog čvora
 - ▶ $key > k$ AND exists next k — $k = \text{next } k$

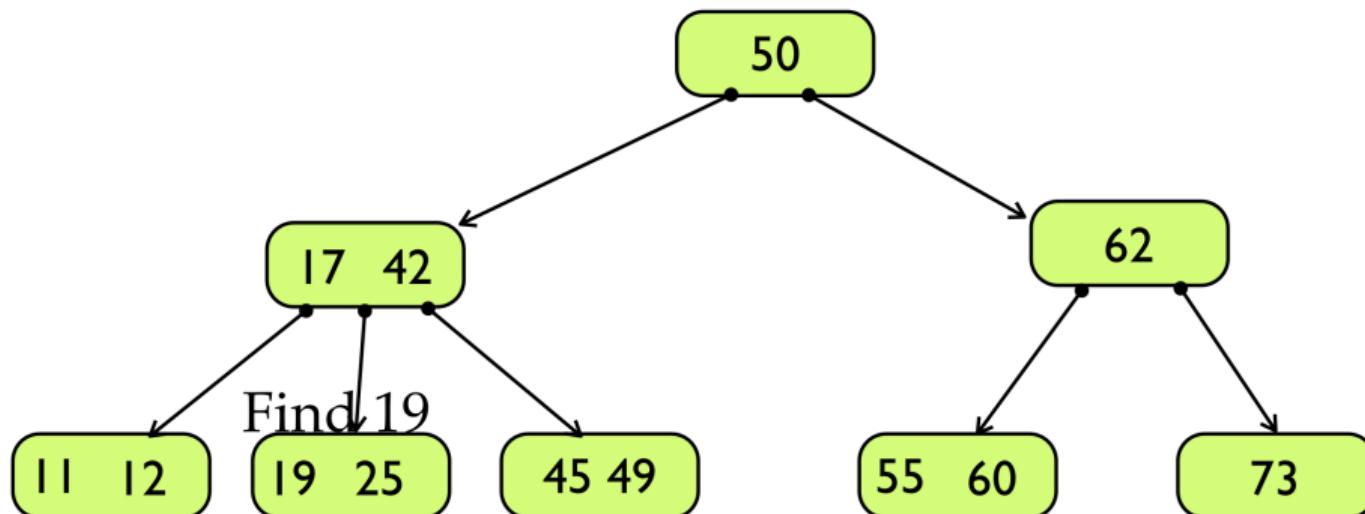
Primer



Primer



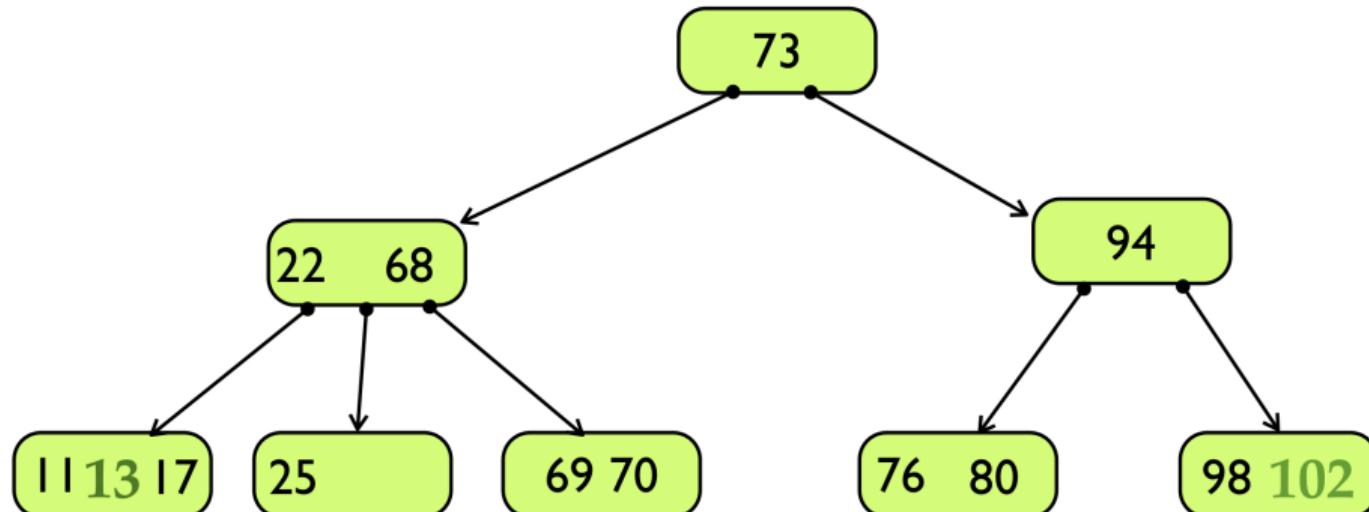
Primer



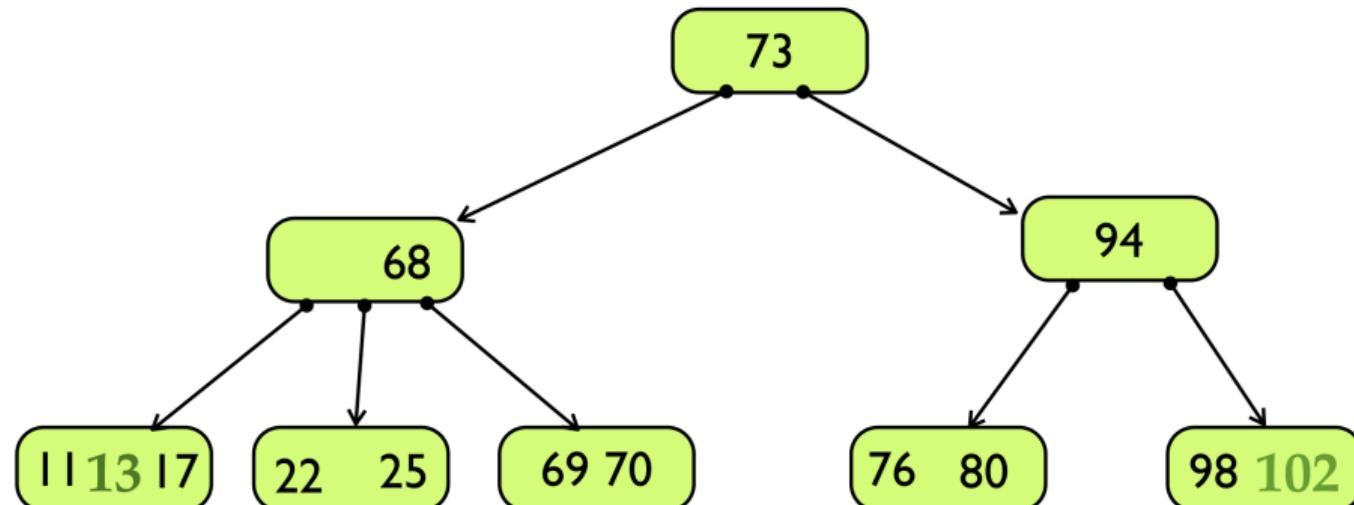
Dodavanje

- ▶ Započinje neuspešnom pretragom ključa i uvek se završava u listu
- ▶ Kada dodamo ključ u odgovarajući čvor i on nema više od MAX elemenata, završavamo proceduru
- ▶ U suprotnom, radimo rotaciju ključeva:
 - ▶ Pronađi sibling čvor koji sadrži manje od MAX elemenata
 - ▶ Prebaci odgovarajući ključ iz roditelja u sibling čvor
 - ▶ Prebaci novi ključ iz deteta u roditeljski čvor
- ▶ Ili podelu čvora (kada rotacija nije moguća):
 - ▶ Čvor delimo na tri dela: ključ u sredini, ključeve pre njega i ključeve posle njega
 - ▶ Ključ iz sredine prebacujemo u roditeljski čvor
 - ▶ Od preostala dva niza ključeva pravimo dva nova čvora, na koje će pokazivati roditeljski čvor
 - ▶ **Napomena** — prebacivanje ključeva može dovesti do overflow-a u roditeljskim čvorovima, rekurzivno delimo čvorove sve do korena

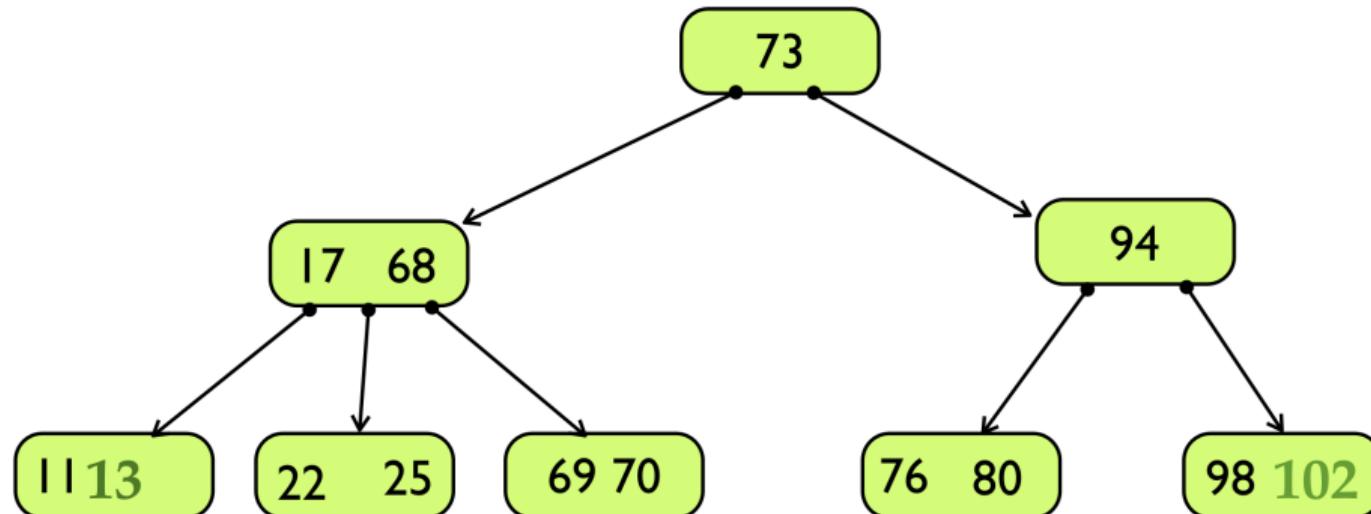
Primer - Rotacija



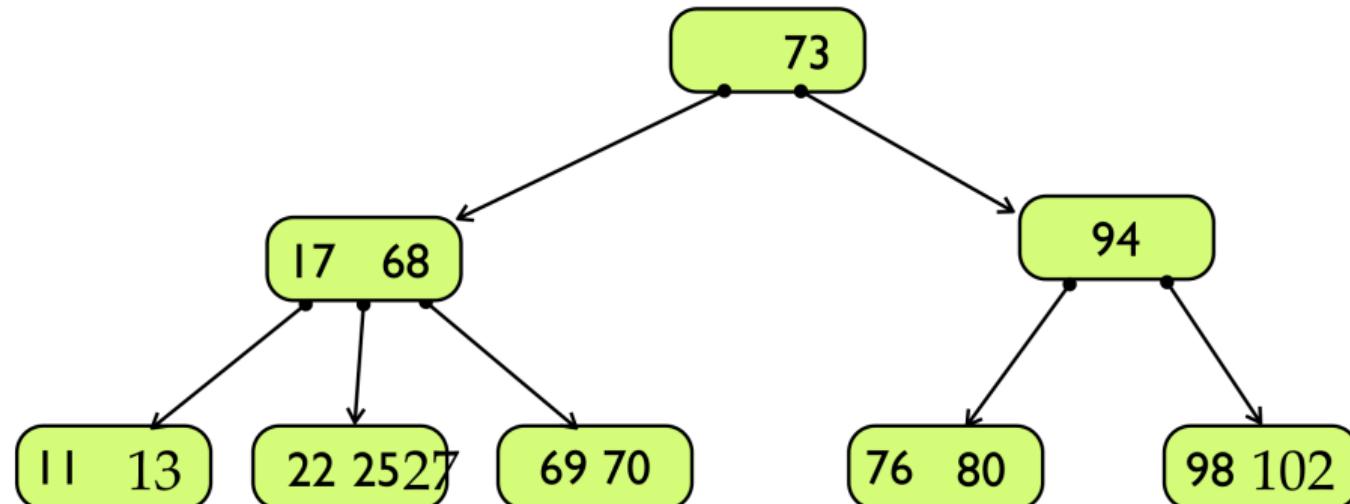
Primer - Rotacija



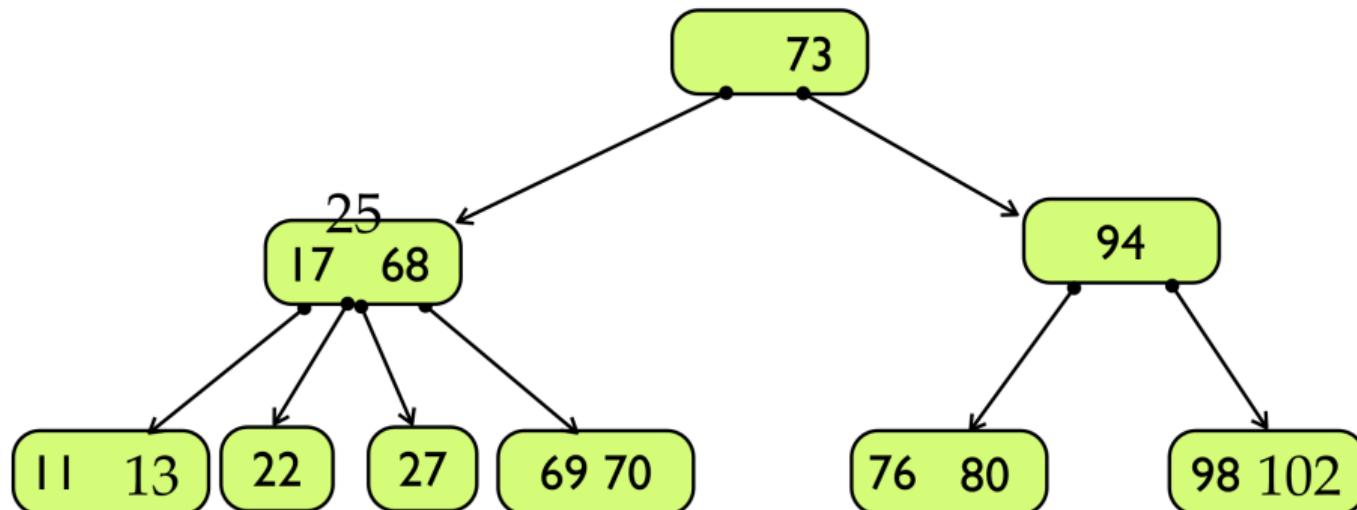
Primer - Rotacija



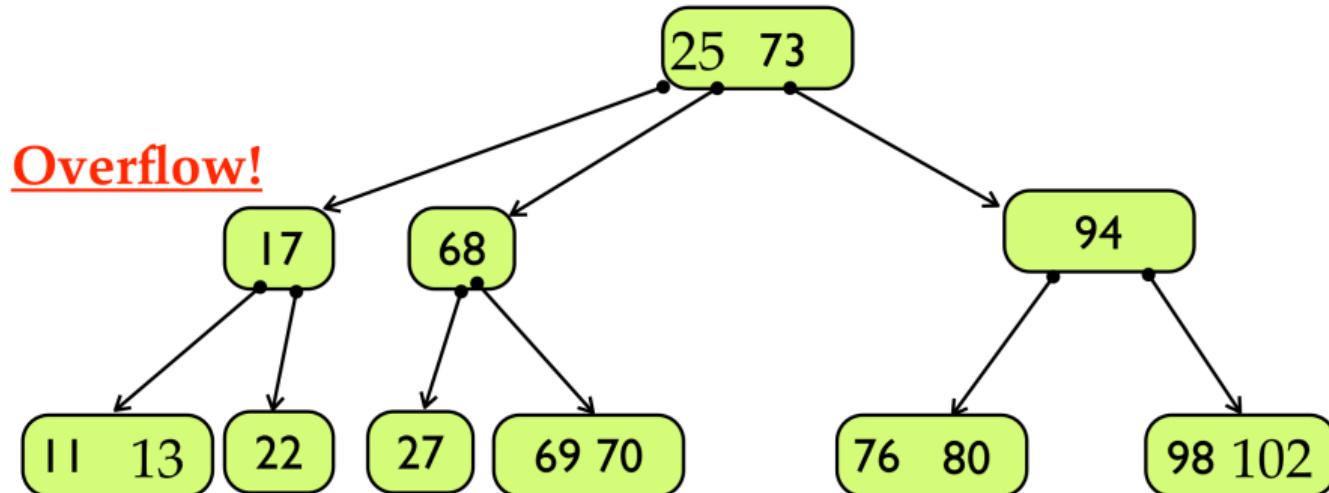
Primer - Podela čvorova



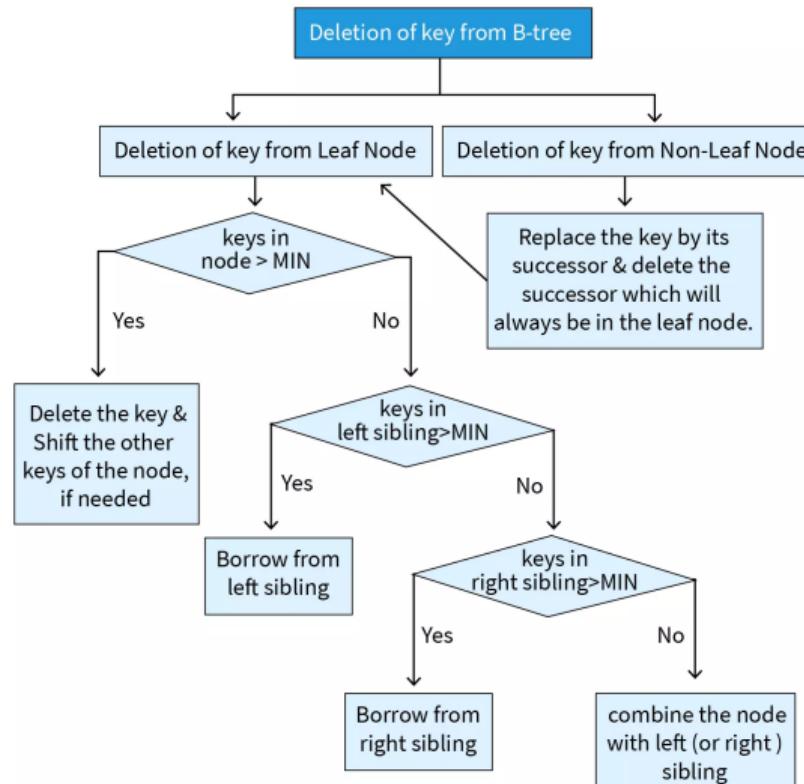
Primer - Podela čvorova



Primer - Podela čvorova

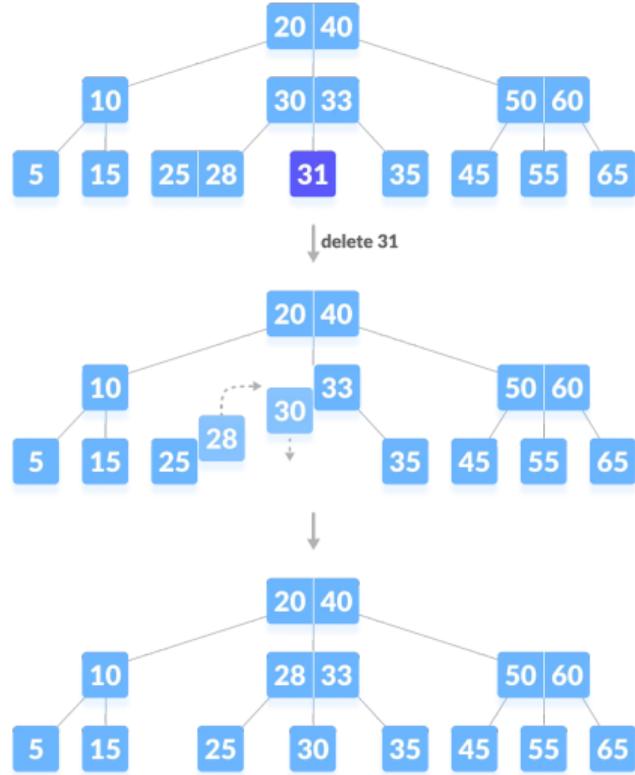


Brisanje



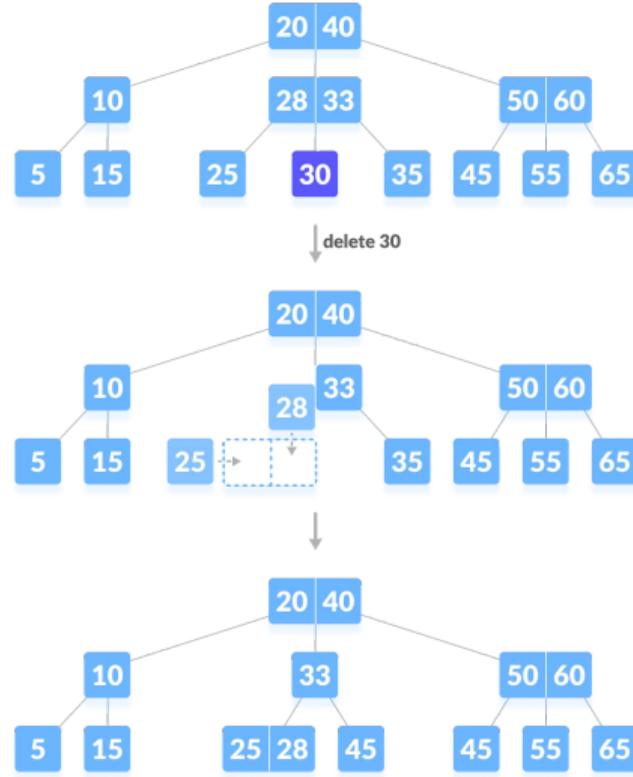
Pozajmljivanje iz sibling čvora

- ▶ Kada uzimamo iz levog sibling-a:
 - ▶ Preuzimamo ključ iz roditeljskog čvora koji je prethodnik čvora obrisanog ključa
 - ▶ Na ispraznjeno mesto u roditeljskom čvoru dodajemo najveći ključ iz levog sibling-a
- ▶ Kada uzimamo iz desnog sibling-a:
 - ▶ Preuzimamo ključ iz roditeljskog čvora koji je sledbenik čvora obrisanog ključa
 - ▶ Na ispraznjeno mesto u roditeljskom čvoru dodajemo najmanji ključ iz desnog sibling-a
- ▶ **Napomena 1:** sibling se ne mora nalaziti neposredno pored čvora i u tom slučaju pozajmljivanje se obavlja u više koraka



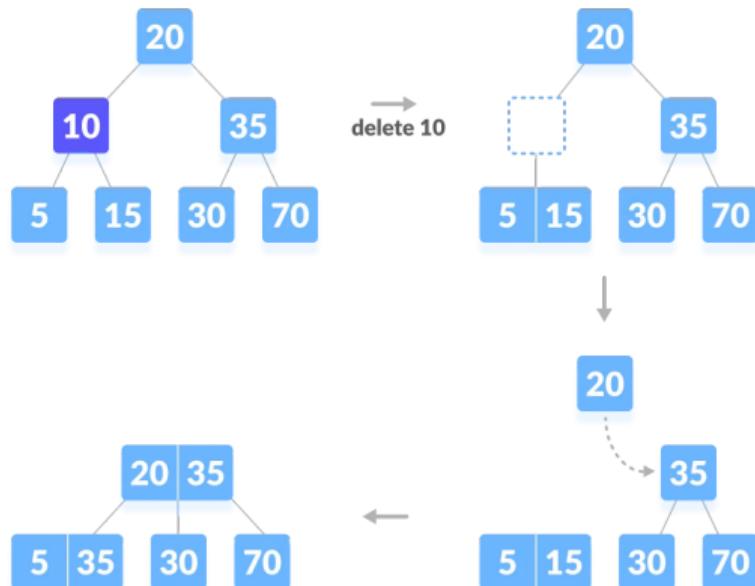
Spajanje čvorova

- ▶ Strategiju koristimo kada i levi i desni sibling čvorovi imaju minimalan broj ključeva
- ▶ Ključ iz roditeljskog čvora, koji je prethodnik ili sledbenik čvora obrisanog ključa, prebacujemo na poslednje mesto levog čvora
- ▶ Sve elemente desnog čvora dodajemo na kraj levog čvora
- ▶ Uklanjamo premešteni ključ iz roditeljskog čvora, kao i sada prazan desni čvor



Spajanje čvorova

- ▶ **Napomena:** Spajanjem čvorova roditelj može spasti na broj ključeva manji od minimuma
- ▶ Moramo rekursivno proći kroz roditeljske čvorove, sve do korena, i popuniti čvorove koji nemaju dovoljno ključeva
- ▶ **Napomena:** ako pozajmljujemo iz sibling-a koji nije list, potrebno je da podstablo koje je ostalo "višak" prebacimo u čvor koji je imao manjka ključeva



Dodatni materijali

- ▶ Organization and maintenance of large ordered indices
- ▶ The Ubiquitous B-Tree
- ▶ B-Tree Visualization
- ▶ 2-3 Trees and B-Trees
- ▶ B+Tree Indexes

Zadaci

- ▶ Implementirati B stablo strukturu podataka
- ▶ Podržati operacije pretrage, dodavanja i brisanja elemenata
- ▶ Dodati funkciju koja vraća listu svih elemenata stabla, sortiranih u rastućem redosledu