

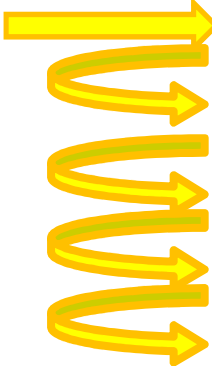
FIZIČKA ORGANIZACIJA PODATAKA

XII predavanje

Dr.sc. Emir Mešković

Metode pohrane i adresiranja podataka u vanjskoj memoriji

- ▶ Različite metode adresiranja koriste se za pronalaženje fizičke pozicije zapisa (ili fizičke pozicije bloka u kojem se taj zapis nalazi) na osnovu vrijednosti ključa (primarnog, alternativnog, "sekundarnog ključa")
- ▶ Sekvencijalna pretraga "heap" organizovane datoteke
 - ▶ najjednostavnija metoda pohrane i adresiranja
 - ▶ podaci ne trebaju biti poredani prema ključu prema kojem se obavlja pretraga



15	Damir
1	Maja
47	Goran
2	Ema
5	Dino

- N_F broj zapisa
- N_S broj inspekcija
- $E(N_S)$ srednji broj ispitivanja kod traženja zapisa s ključem k

$$E(N_S) = N_F / 2$$

Blokovsko pretraživanje (*block search, skip search*)

1	Damir
3	Maja
5	Goran
9	Ema
11	Dino
13	Lana
17	Deni
19	Bakir
27	Sara
29	Jasna
34	Denis
37	Saša
43	Tea
47	Elvis

- ▶ Podaci moraju biti poredani prema ključu prema kojem se obavlja pretraga
- ▶ Čita se prvi zapis iz svakog bloka sve dok se ne pronade zapis s ključem koji je veći od traženog. Nakon toga se sekvencijalno pretražuje posljednji preskočeni blok.
- ▶ N_F broj zapisa
- ▶ N_S broj čitanja
- ▶ N_B broj zapisa u bloku
- ▶ $E(N_S)$ srednji broj ispitivanja kod traženja zapisa s ključem k
- ▶ $E(N_S) = (N_F / N_B) / 2 + N_B / 2$

Blokovsko pretraživanje (*block search, skip search*)

- ▶ Optimalna veličina bloka:
 - ▶ $E(N_S) = (N_F / N_B) / 2 + N_B / 2$
 - ▶ deriviranjem po N_B
 - ▶ Izjednačavanjem s nulom
- ▶ Optimalna veličina bloka: $N_B = N_F^{1/2}$
- ▶ Srednji broj inspekcija korištenjem optimalne veličine bloka
 - ▶ $E(N_S) = N_F^{1/2}$



Binarno pretraživanje (*binary search*)

1	Damir
3	Maja
5	Goran
9	Ema
11	Dino
13	Lana
17	Deni
19	Bakir
27	Sara
29	Jasna
34	Denis
37	Saša
43	Tea
47	Elvis

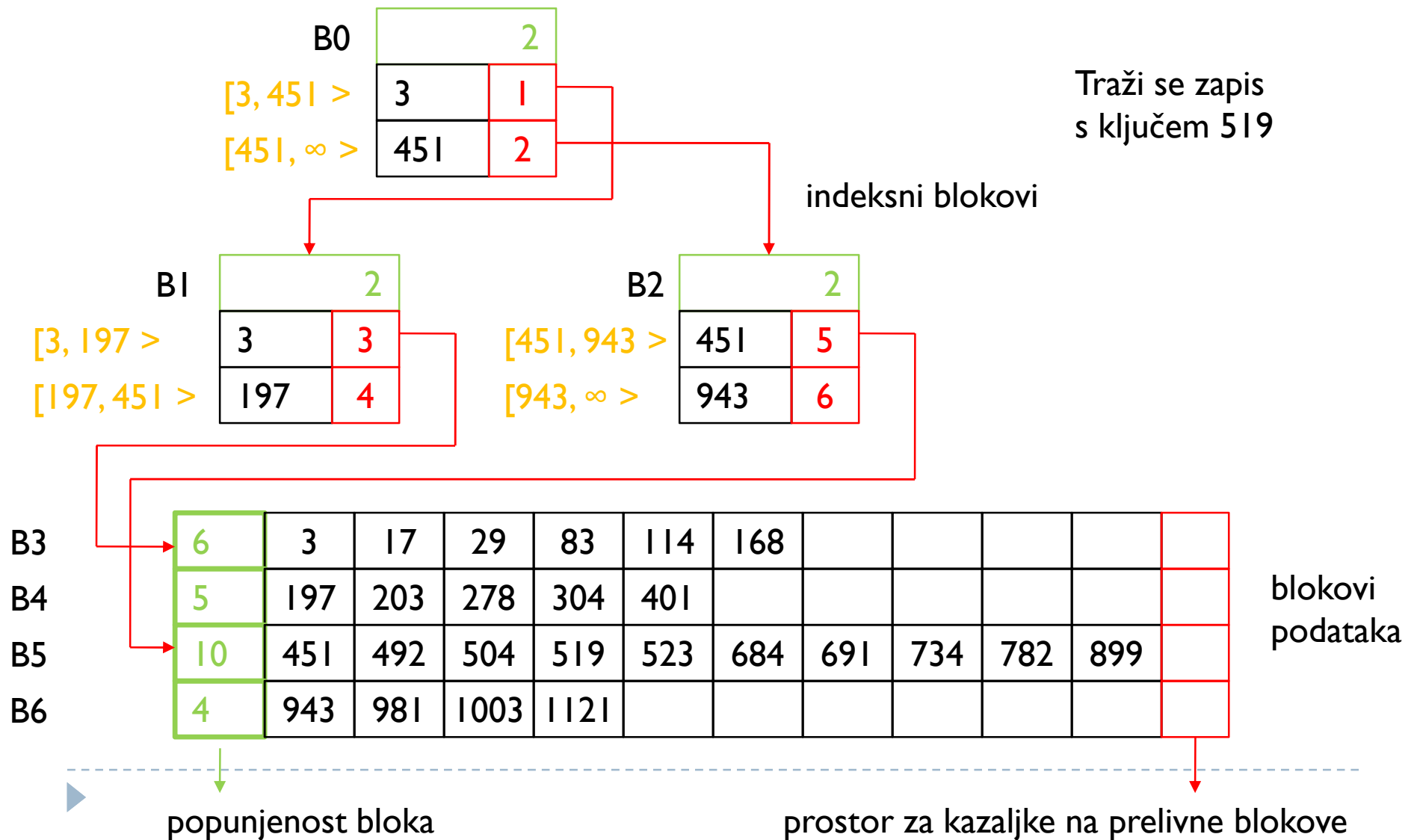
- ▶ Podaci moraju biti poredani prema ključu prema kojem se obavlja pretraga
- ▶ Područje pretrage se uvijek dijeli na pola. Ovisno o vrijednosti ključa koja je pronađena na mjestu preloma, pretraga se nastavlja u segmentu ispod ili iznad preloma.
- ▶ NF broj zapisa
- ▶ NS broj čitanja
- ▶ Broj ispitivanja u najgorem slučaju:
 - ▶ $NS = \log_2 NF + 1$
- ▶ $E(NS)$ srednji broj ispitivanja kod traženja zapisa s ključem k
 - ▶ $E(NS) \approx \log_2 NF - 1$

Indeksno sekvencijalna organizacija (*Index-Sequential Organization*)

- ▶ podaci su poredani prema ključu po kojem se obavlja pretraga
- ▶ podaci su podijeljeni u blokove koji se mogu obrađivati sekvencijalno
- ▶ stvara se kazalo na nekoliko nivoa
- ▶ INDEKSNI BLOKOVI
- ▶ BLOKOVI PODATAKA
- ▶ zapis indeksnog bloka sadrži najmanju vrijednost ključa u podređenom bloku i kazaljku prema podređenom bloku



Indeksno sekvencijalna organizacija



B3	6	3	17	29	83	114	168					
B4	5	197	203	278	304	401						
B5	10	451	492	504	519	523	684	691	734	782	899	
B6	4	943	981	1003	1121							

Dodavanje zapisa s ključem 219

B4	6	197	203	219	278	304	401					
----	---	-----	-----	-----	-----	-----	-----	--	--	--	--	--

Dodavanje zapisa s ključem 743

U bloku B5 više nema mjesta pa se otvara novi podatkovni blok u prelivnom području

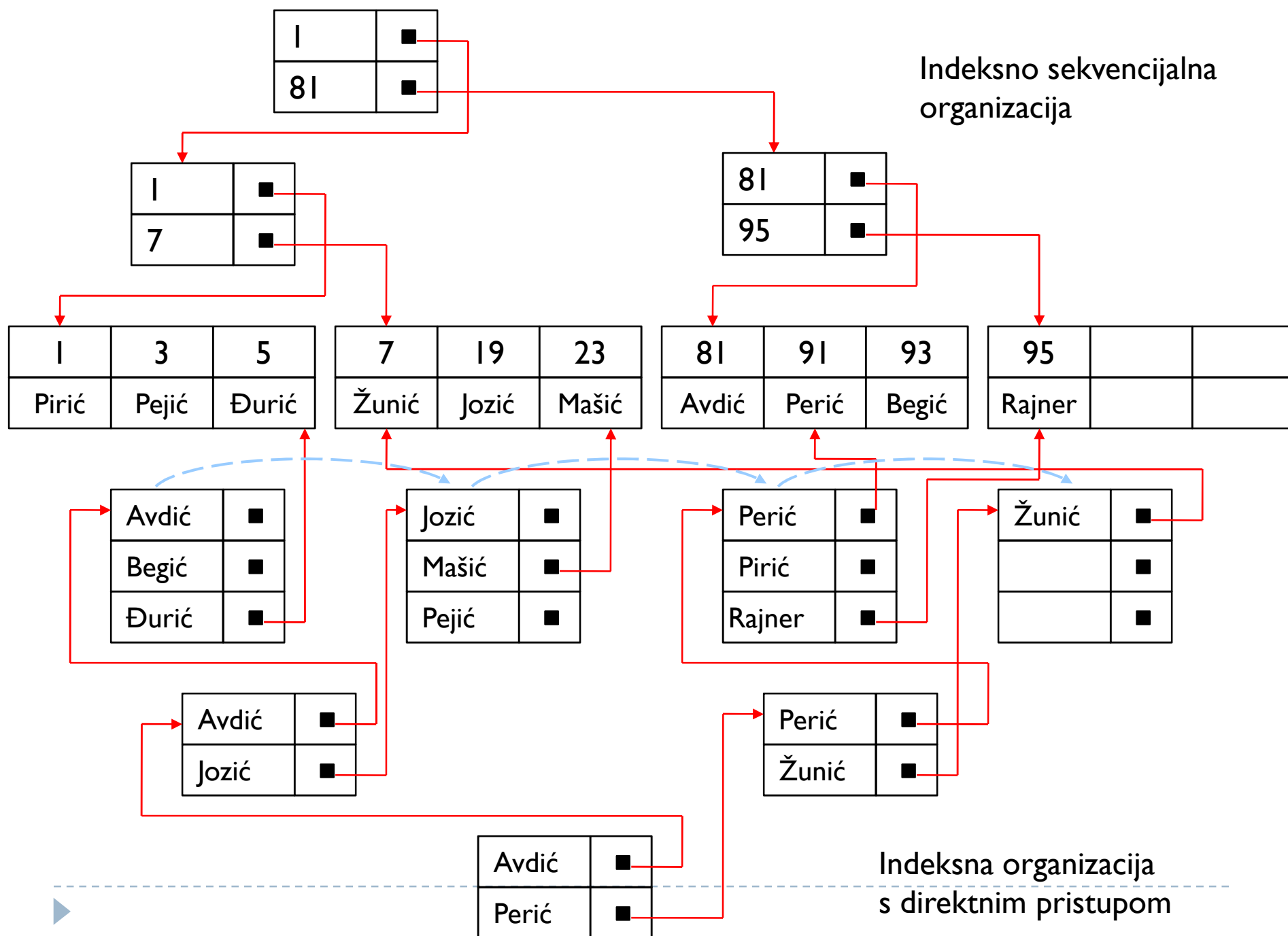
B5	10	451	492	504	519	523	684	691	734	743	782	
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	--

[illegible]

Indeksno sekvencijalna organizacija

- ▶ kad prelivno područje postane preveliko, pretraživanje postaje neefikasno -potrebno je obaviti reorganizaciju
- ▶ brisanje je najčešće logičko -zapis dobija oznaku da je obrisano, a fizički se uklanja tek kod reorganizacije
- ▶ Indeksna organizacija s direktnim pristupom
 - ▶ šta ako podaci nisu poredani prema ključu?
 - ▶ podaci npr. ne mogu biti poredani prema ključu ako se zahtijeva pristup prema više nego jednom ključu – ako su podaci poredani prema jednom ključu, tada ne mogu biti poredani prema nekom drugom ključu
 - ▶ ukoliko zapisi nisu poredani prema ključu potrebno je izgraditi indeks koji sadrži indeksni zapis za svaki zapis podataka, umjesto na svaki blok podataka kao što je to u indeksno sekvencijalnoj organizaciji
 - ▶ blokovi posljednjeg indeksnog nivoa mogu biti povezani kazaljka da bi se omogućila sekvencijalna obrada zapisa s vrijednostima ključa





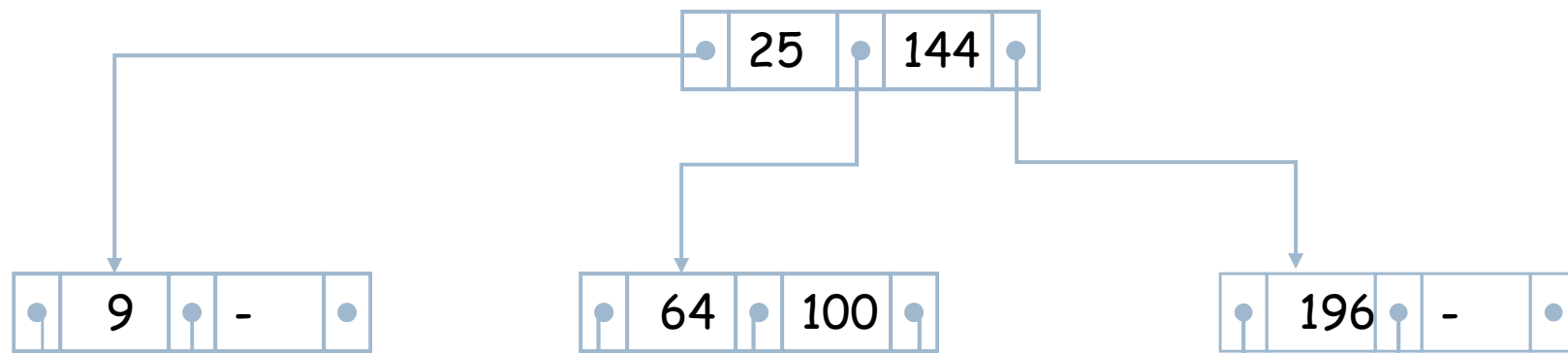
B stablo (balansirano stablo)

- ▶ Organizacija B-stabla (B-tree, Balanced Tree) slična je indeksno sekvencijalnoj organizaciji
- ▶ Datoteka B-stabla također se sastoji od nekoliko nivoa indeksnih blokova i podatkovnih blokova
- ▶ Osnovna razlika između indeksno sekvencijalne organizacije i B-stabla je u tome što se kod B-stabla broj nivoa indeksa dinamički mijenja pa ne postoji potreba za reorganizacijom indeksnih blokova.



B stablo (balansirano stablo)

Indeksni blokovi



Blokovi podataka

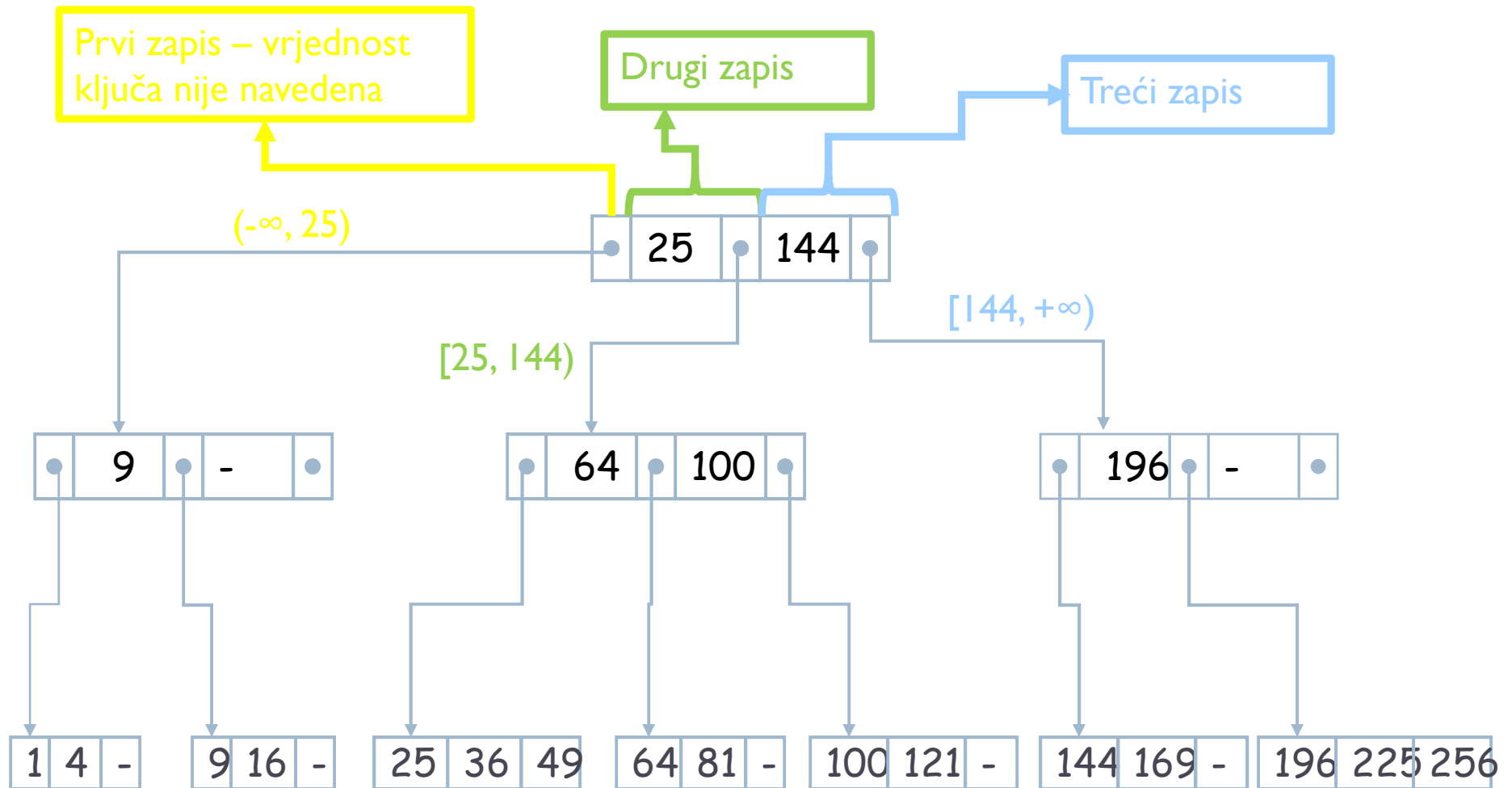


B stablo (balansirano stablo)

- ▶ Broj indeksnih nivoa se dinamički mijenja
- ▶ Popunjenost svakog bloka (osim korijena) mora biti barem 50 %
- ▶ Broj zapisa u indeksnom bloku
 - ▶ neparni cijeli broj veći ili jednak 3:
 - ▶ broj zapisa = $2 * d - 1 \geq 3, d \in \mathbb{N}$
- ▶ Broj zapisa u podatkovnom bloku
 - ▶ neparni cijeli broj veći ili jednak 3:
 - ▶ broj zapisa = $2 * e - 1 \geq 3, e \in \mathbb{N}$



B stablo (balansirano stablo)

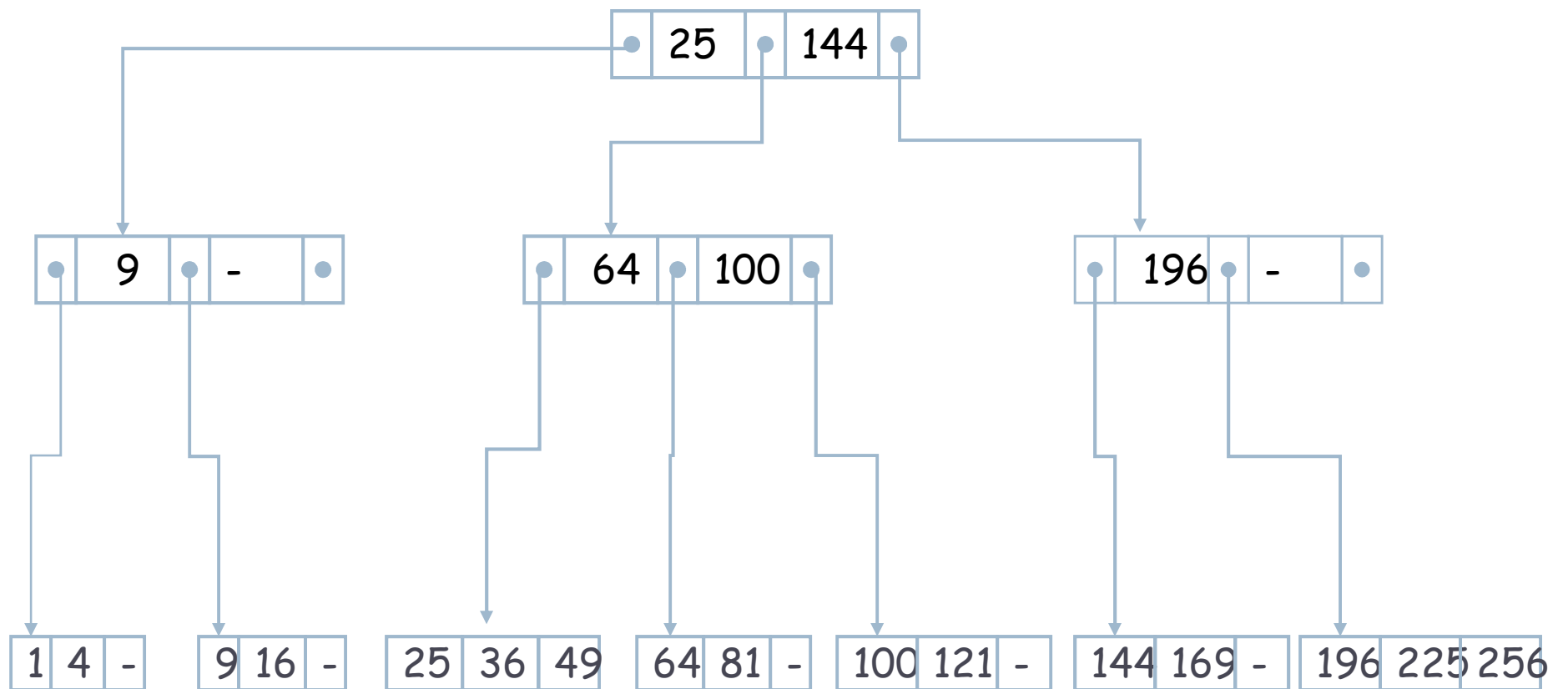


Pretraživanje B-stabla

- ▶ Traženje zapisa s ključem k započinje od korijena (0-tog nivoa)
- ▶ Procedura za traženje je rekurzivna
- ▶ Cilj je u svakom koraku rekurzije (pretraga i -tog nivoa) pronaći blok indeksa (na nižem, $(i+1)$ -vom nivou) koji će voditi prema podatkovnom bloku u kojem se nalazi zapis s ključem k
- ▶ U indeksnom bloku i -tog nivoa potrebno je pronaći najveću vrijednost ključa koja je manja ili jednaka vrijednosti traženog ključa k
 - ▶ za prvu kazaljku u bloku nema navedene vrijednosti ključa pa ona "pokriva" sve vrijednosti ključeva manje od druge vrijednosti ključa u bloku
- ▶ Nakon pronalaženja odgovarajuće vrijednosti ključa, slijedi se pripadna kazaljka i time se obavlja pozicioniranje na $(i+1)$ -vi nivo
- ▶ Postupak se ponavlja rekurzivno sve dok se ne dođe do podatkovnog bloka. U njemu se mora nalaziti, ukoliko postoji, zapis sa ključem k .



Traženje zapisa s
ključem 9, 81, 225

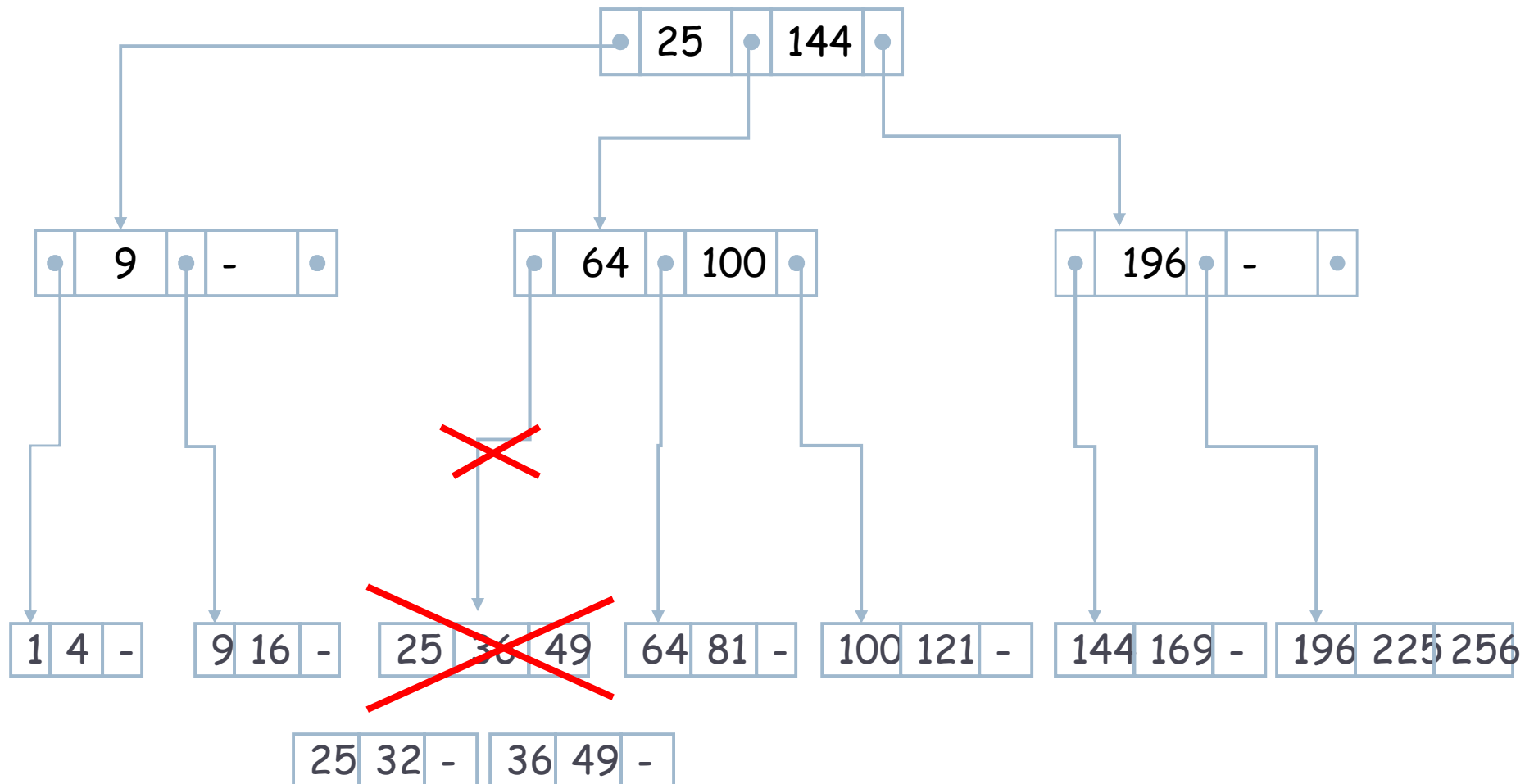


Dodavanje novih zapisa u B-stablo

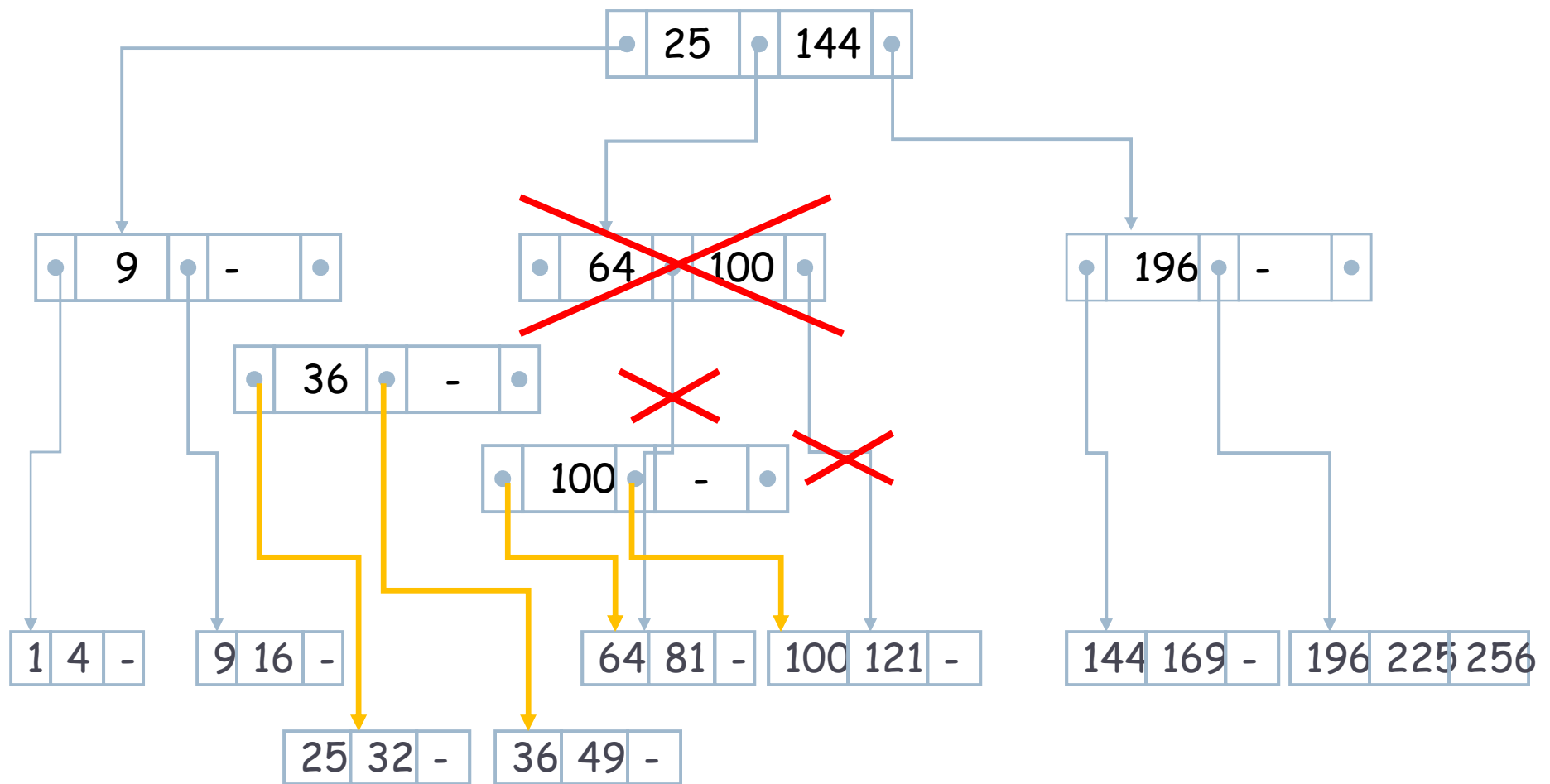
- ▶ Obavlja se procedura za pronalaženje podatkovnog bloka u koji pripada novi zapis s vrijednošću ključa k
- ▶ Ako u bloku postoji manje od $2^*e - 1$ zapisa, novi zapis se dodaje u blok na odgovarajuće mjesto, tako da zapisi budu poredani prema vrijednosti ključa
- ▶ Novi zapis ne može nikada biti prvi u bloku, osim u slučaju krajnjeg lijevog lista
- ▶ Ako je blok u koji treba dodati novi zapis popunjen
 - ▶ Stvara se novi blok i zapisi se među njima podijele, pri čemu svakom bloku pripadne e zapisa
 - ▶ Budući da je dodan novi blok podataka, u nadređeni indeksni blok potrebno je dodati zapis sa najmanjom vrijednošću ključa u tom novom bloku
 - ▶ Za dodavanje novog zapisa u indeksnom bloku koristi se procedura ista kao za dodavanje novog zapisa u podatkovni blok
 - ▶ Postupak je rekurzivan i mora se obaviti za svaki nadređeni indeksni nivo (sve dok s ene dođe do korijena ili se na nekom od indeksnih nivoa nađe dovoljno mjesta za upis vrijednosti ključa bez dodavanja novih indeksnih blokova).
 - ▶ Ako se dođe do korijena, može se desiti da u korijenu nema mjesta za novi zapis. Tada se dodaje novi blok i dodaje se novi korijen na višem nivou



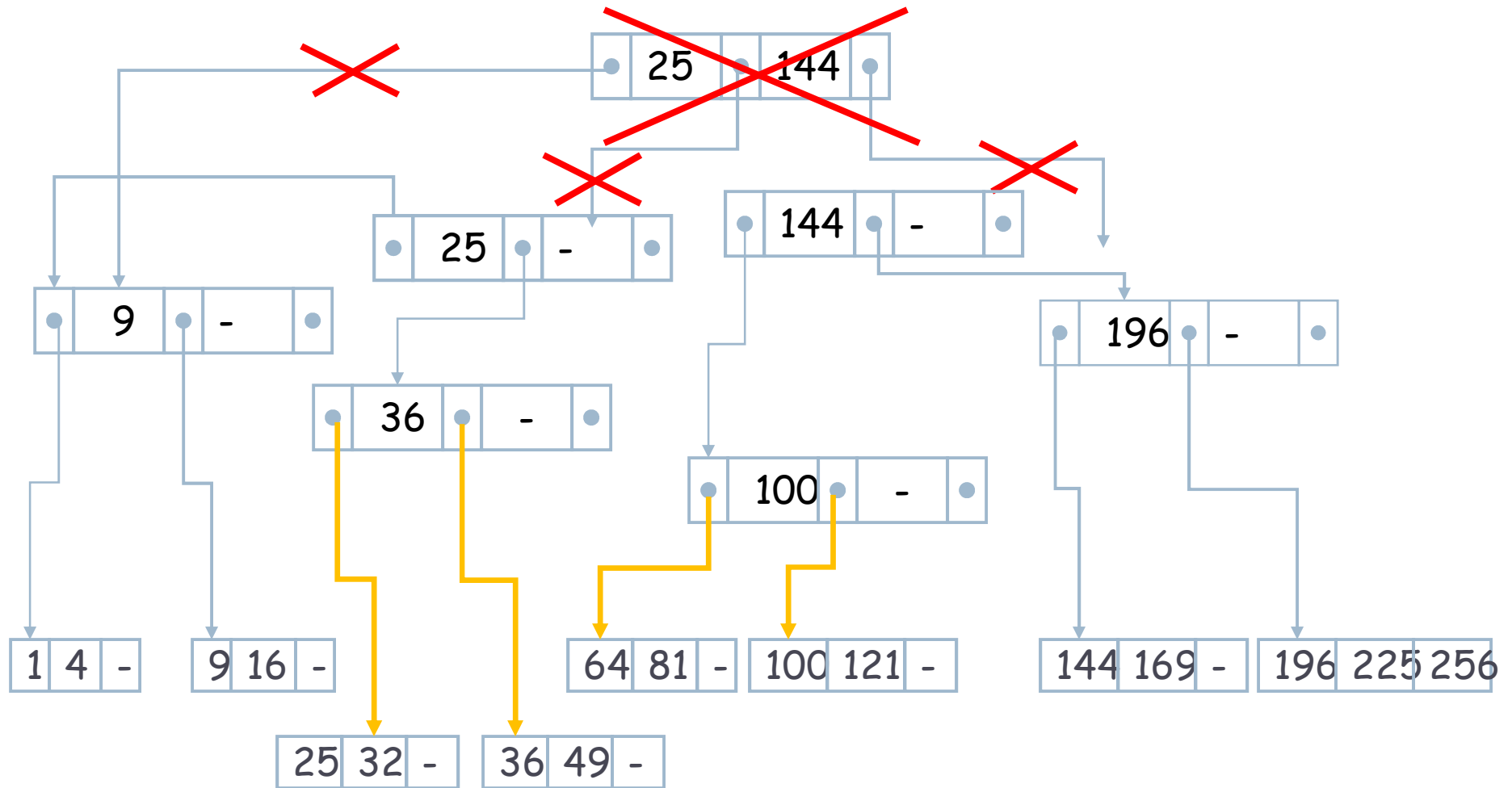
Dodaje se zapis s
ključem 32



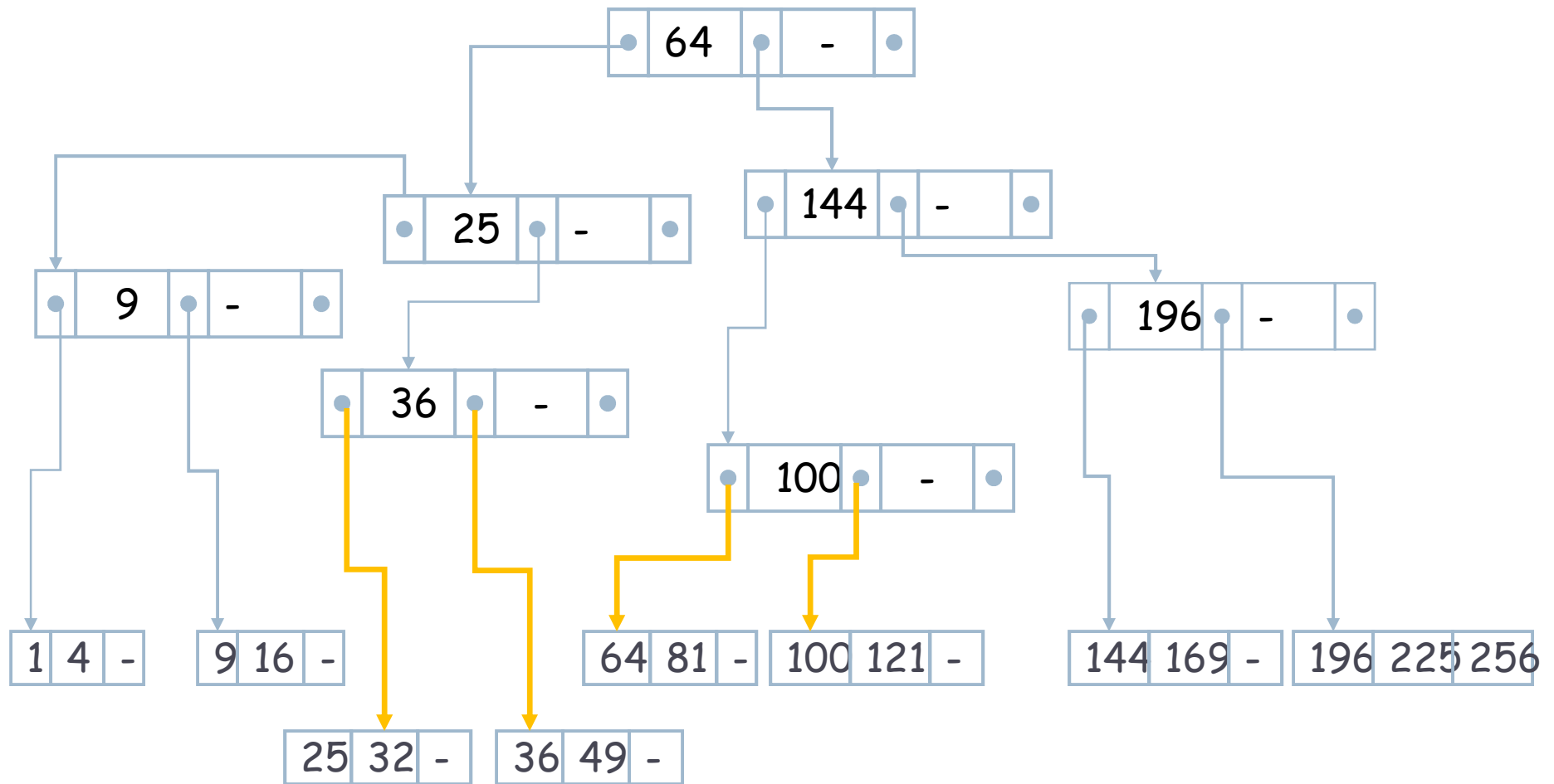
Dodaje se zapis s
ključem 32



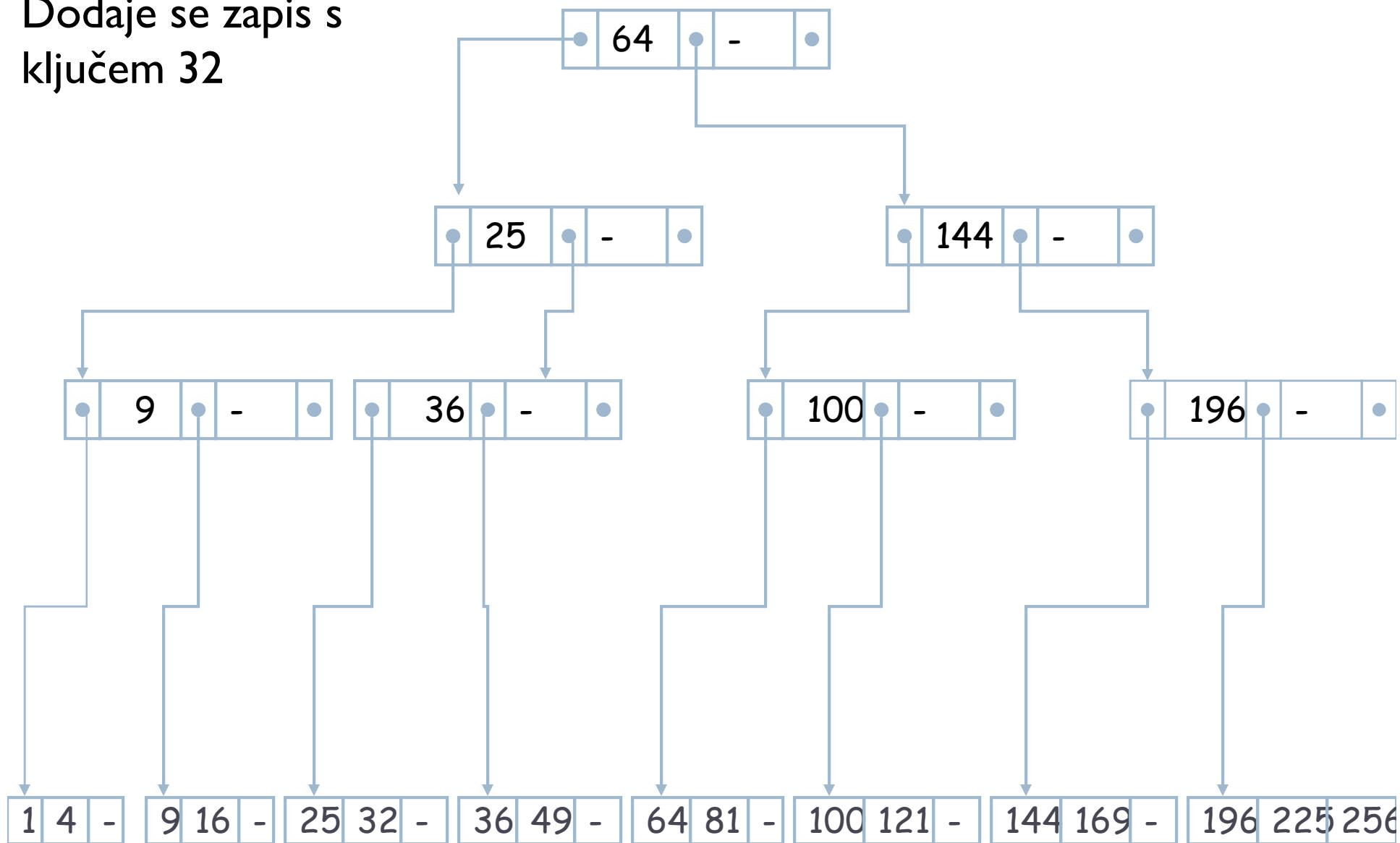
Dodaje se zapis s
ključem 32



Dodaje se zapis s
ključem 32



Dodaje se zapis s
ključem 32

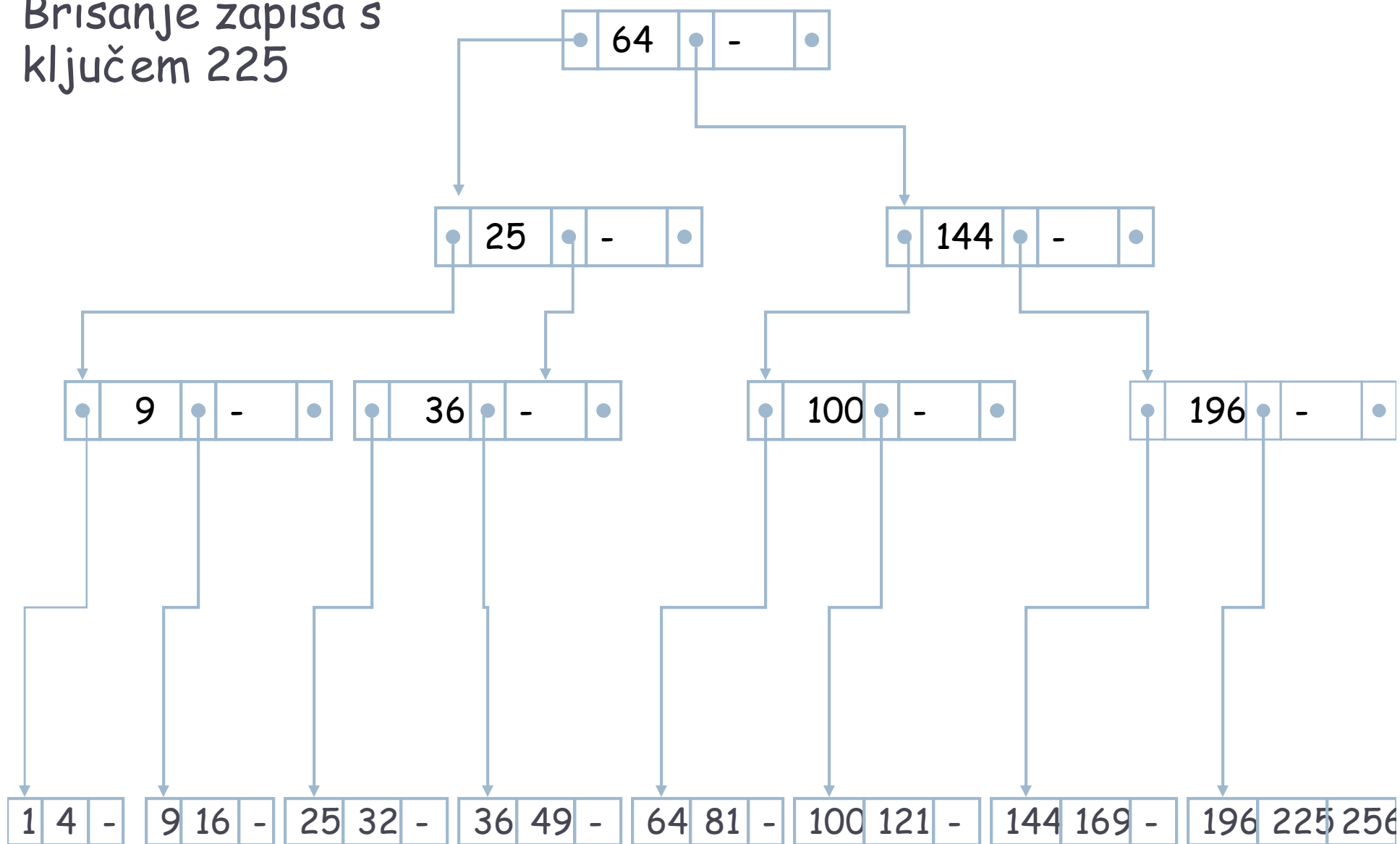


Brisanje zapisa u B-stablu

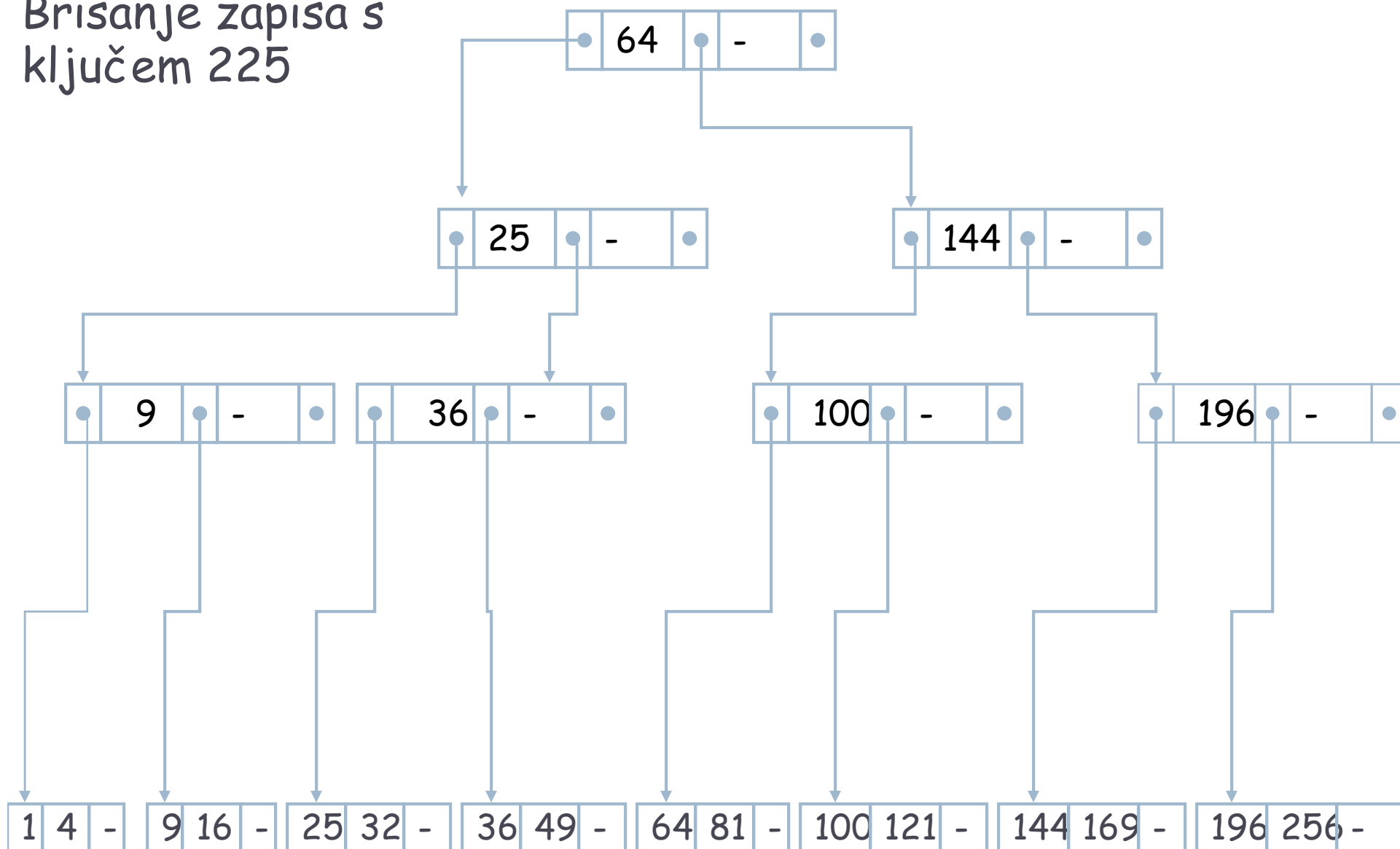
- ▶ Procedurom za pretragu pronalazi se blok u kojem se nalazi zapis kojeg treba obrisati
- ▶ Ako nakon brisanja podatkovni blok sadrži još uvijek barem e zapisa, procedura brisanja se završava uz eventualnu korekciju vrijednosti u nekom od nadređenih indeksnih blokova (ako je obrisani prvi zapis iz bloka)
- ▶ Ako je podatkovni blok B nakon brisanja imao $e-1$ zapis, traži se blok koji ima isti nadređeni indeksni blok i nalazi se odmah s lijeva ili s desna bloku B . Ako odabrani blok B_x ima više od e zapisa, tada se podijele zapisi između blokova B i B_x , uz zadržavanje poretka zapisa u bloku. U nekom od nadređenih indeksnih blokova mora se obaviti i izmjena vrijednosti ključeva.
- ▶ Ako je blok B_x imao upravo e zapisa, tada se spajaju blokovi B i B_x u blok koji će imati upravo $2 \cdot e - 1$ zapisa. U nadređenom bloku mijenja se zapis za B_x i briše zapis za B . Brisanje ovog zapisa svodi se na rekurzivno izvođenje procedure za brisanje. Ako se putem prema korijenu dođe u situaciju da treba spojiti jedina dva bloka – potomka korijena, tada se oni spajaju, postaju korijen stabla i stari korijen se briše.



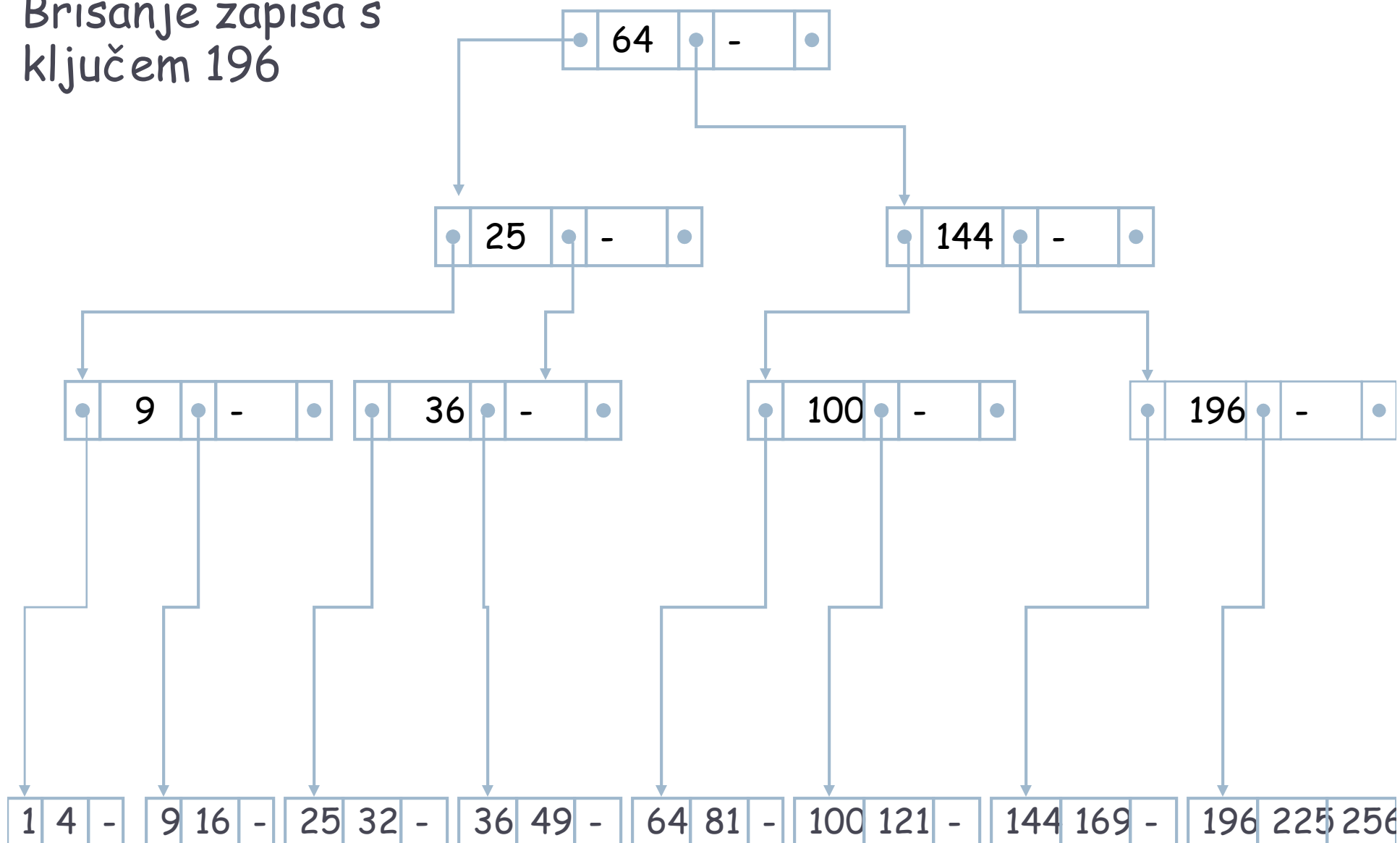
Brisanje zapisa s
ključem 225



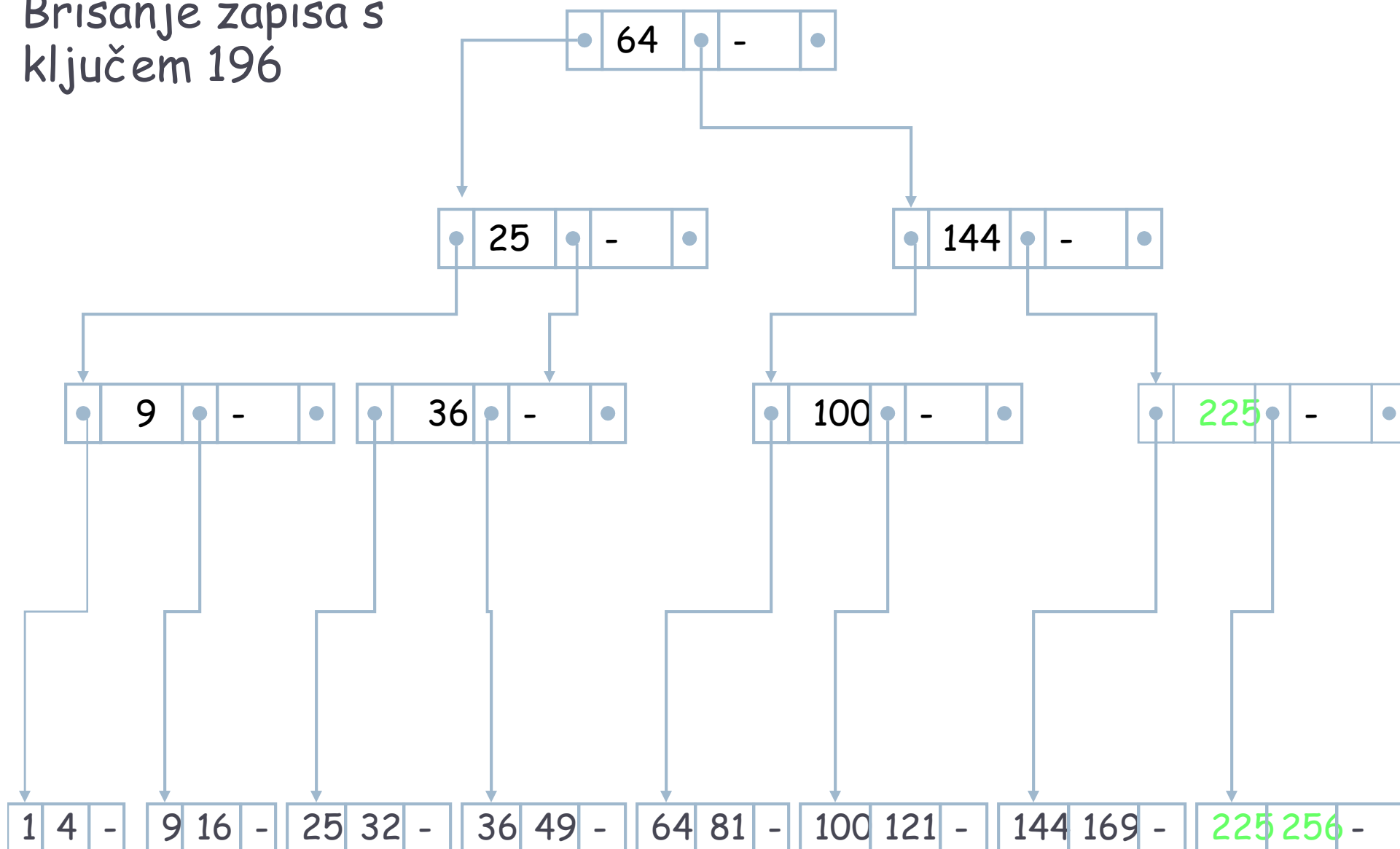
Brisanje zapisa s ključem 225



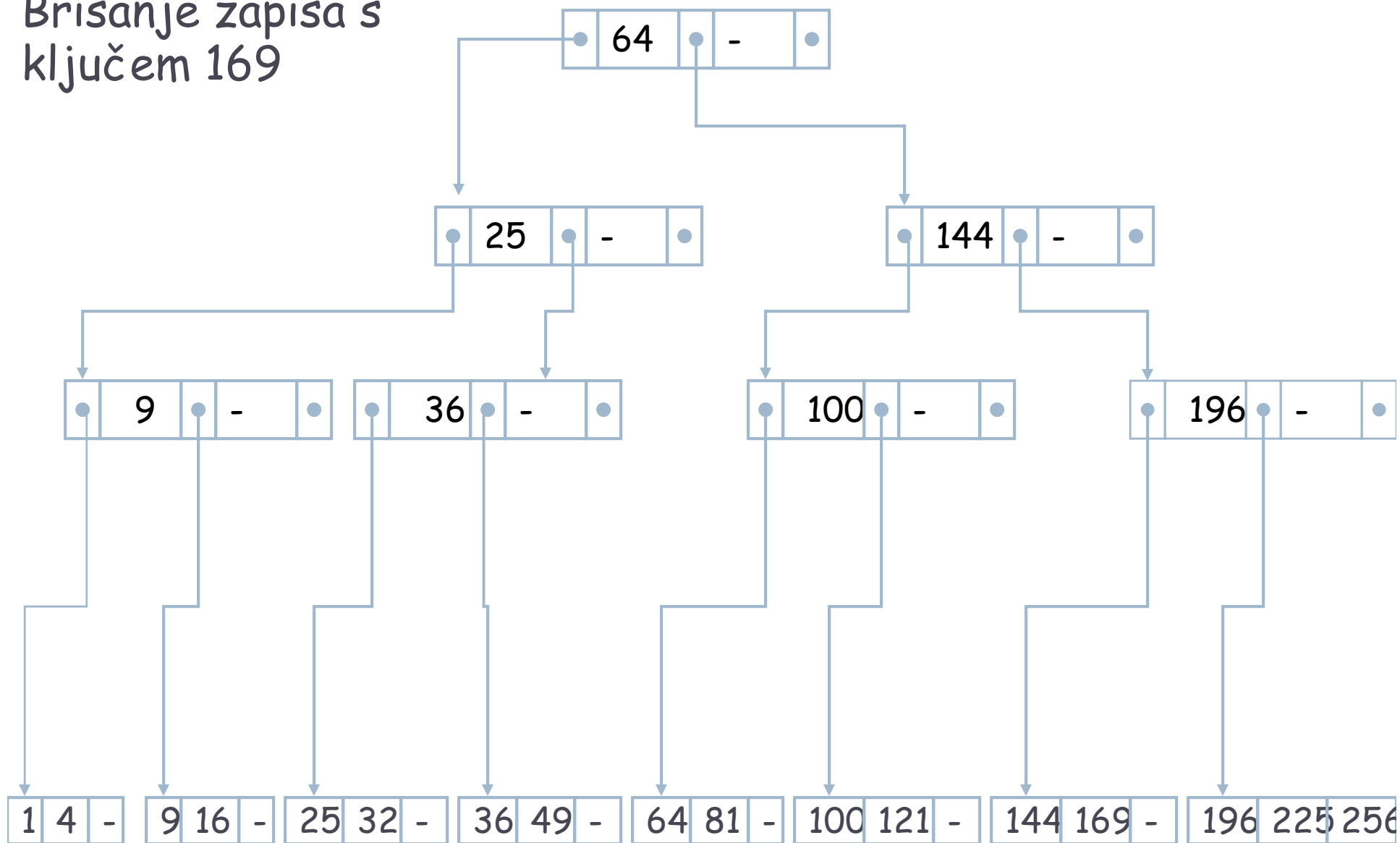
Brisanje zapisa s
ključem 196



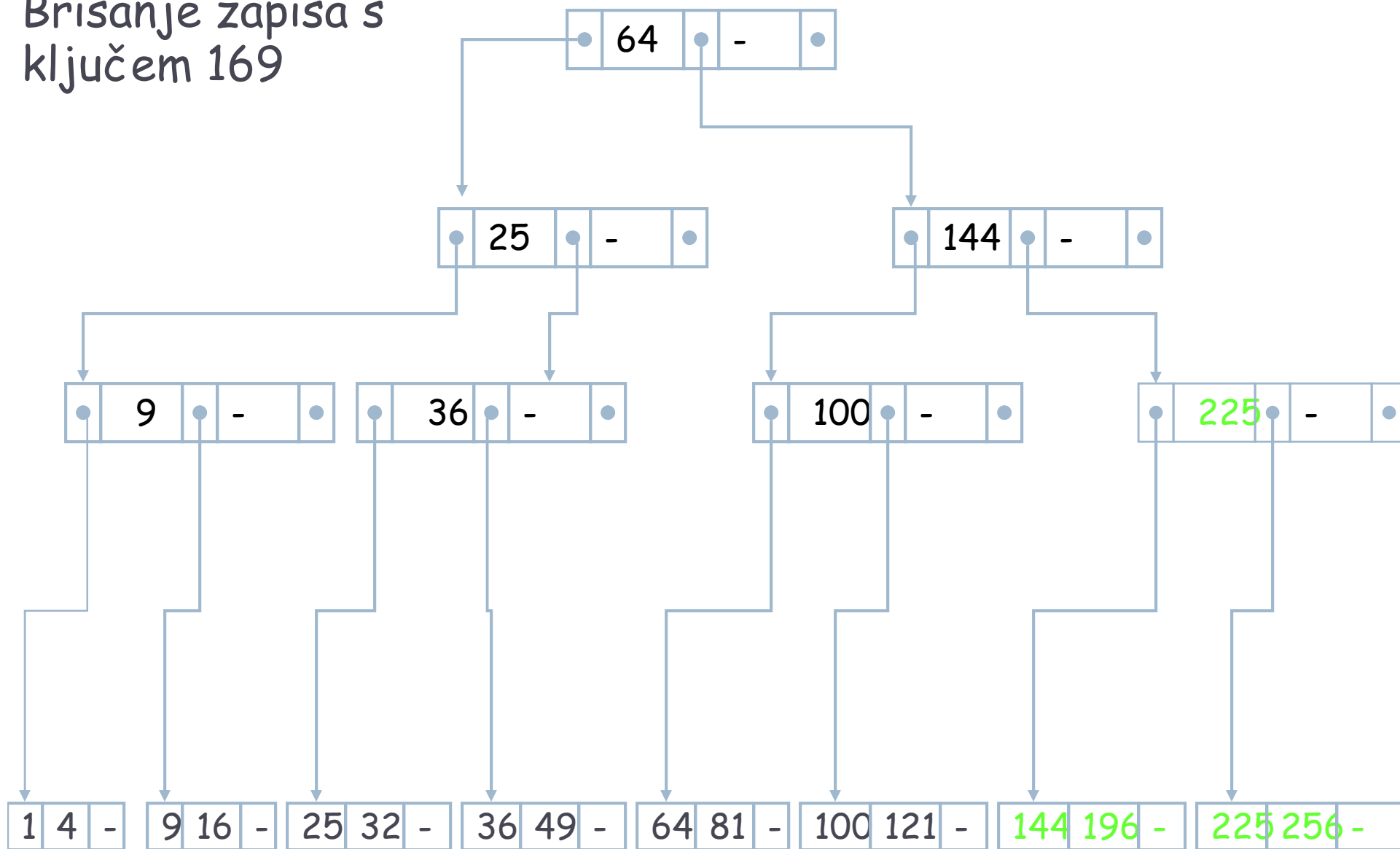
Brisanje zapisa s
ključem 196



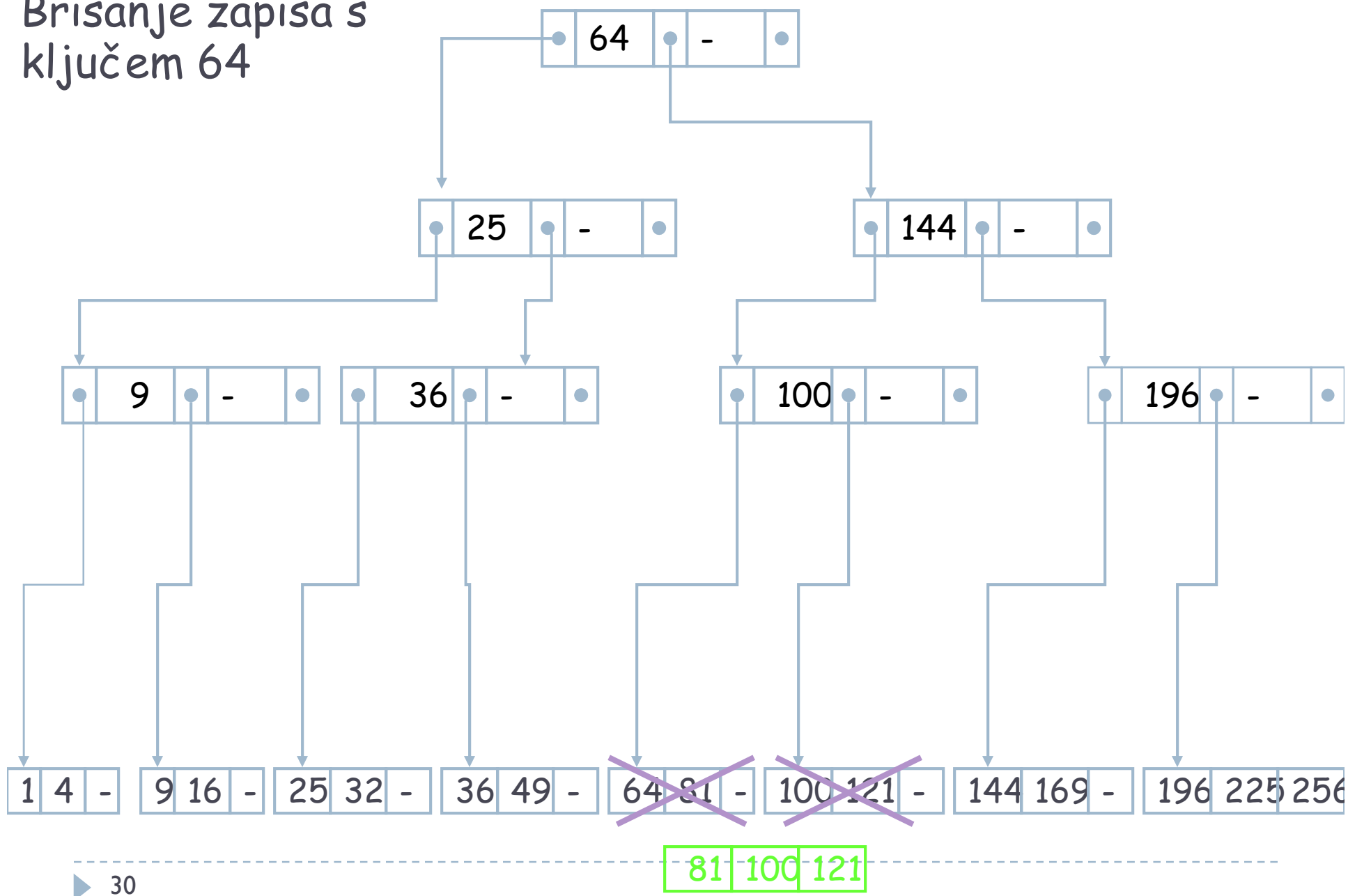
Brisanje zapisa s
ključem 169



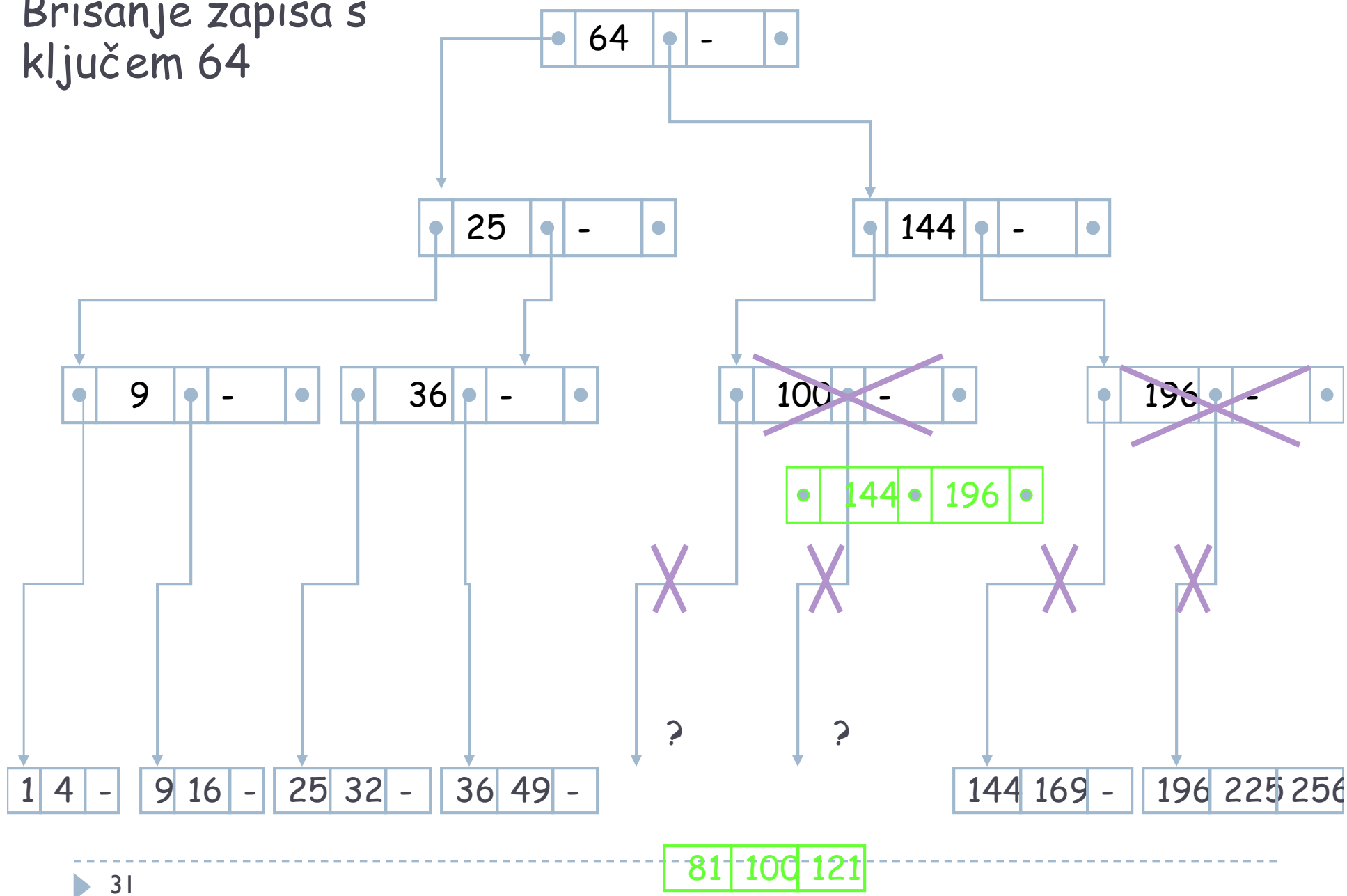
Brisanje zapisa s
ključem 169



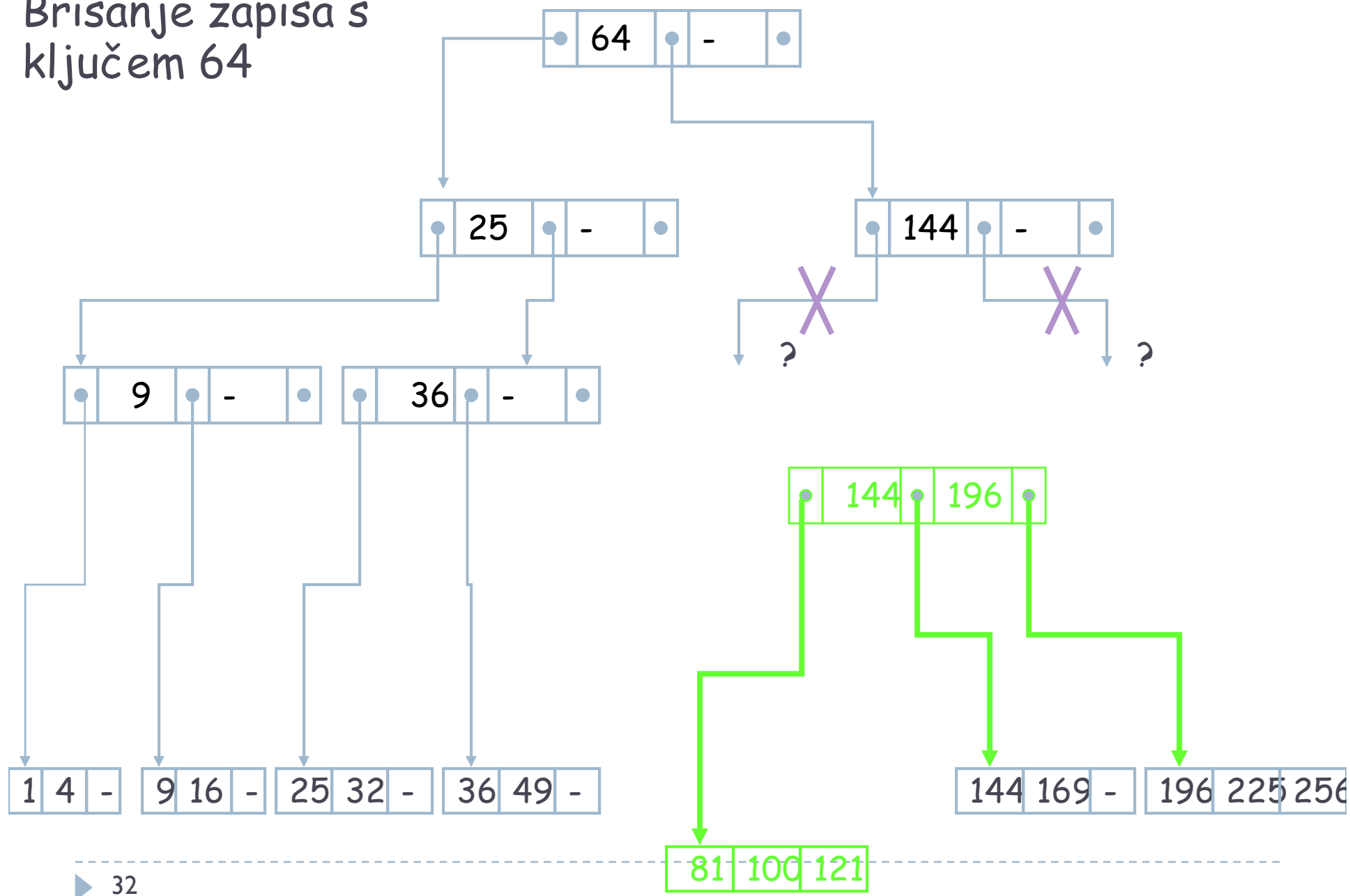
Brisanje zapisa s
ključem 64



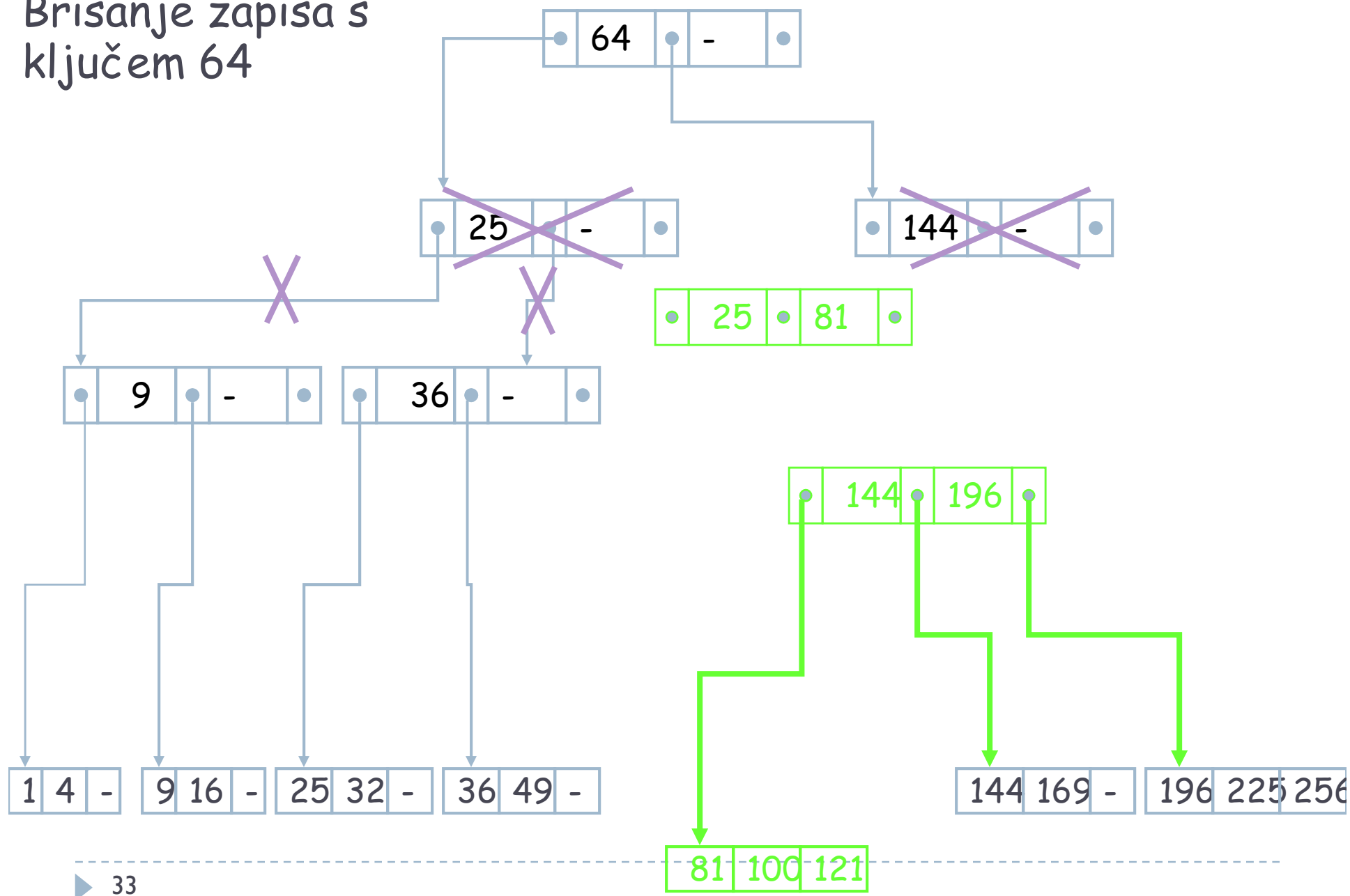
Brisanje zapisa s
ključem 64



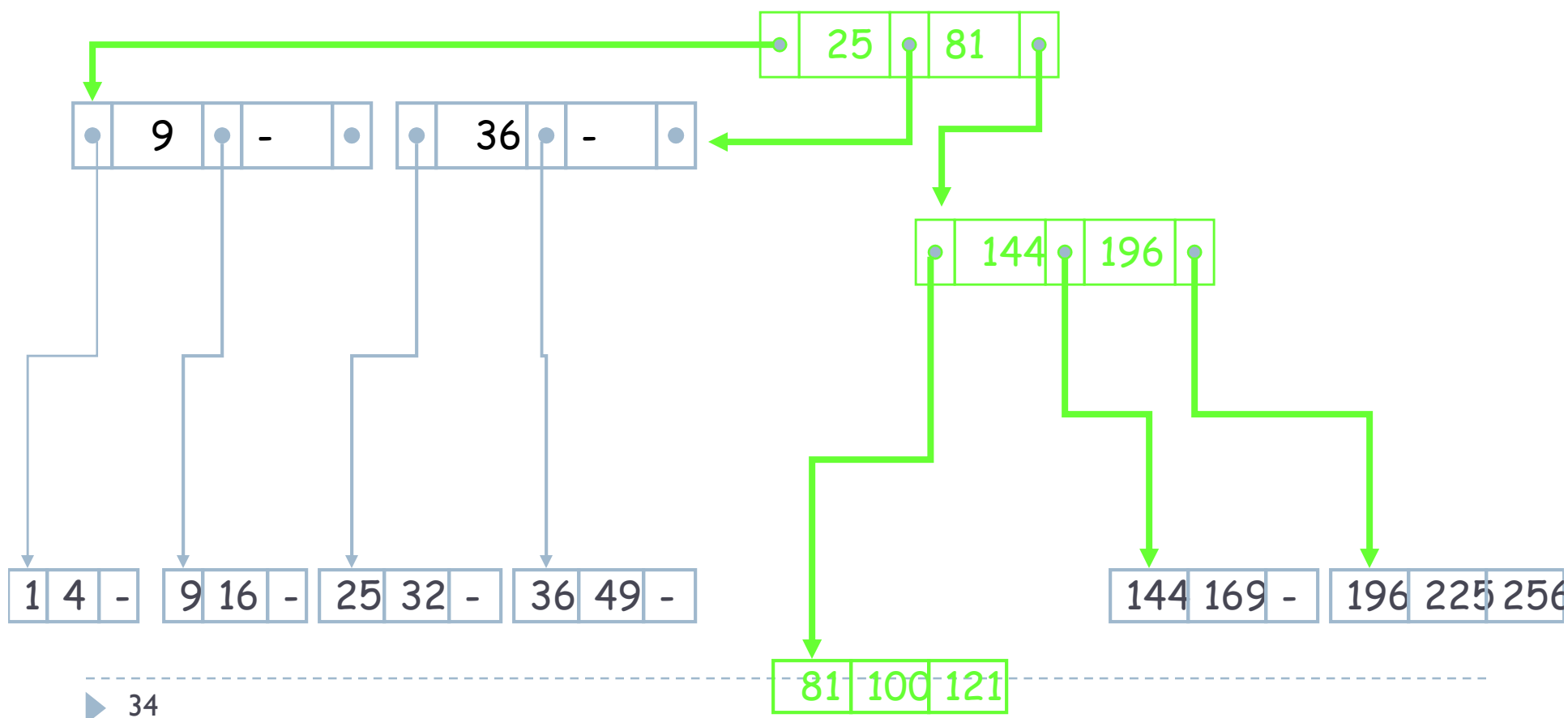
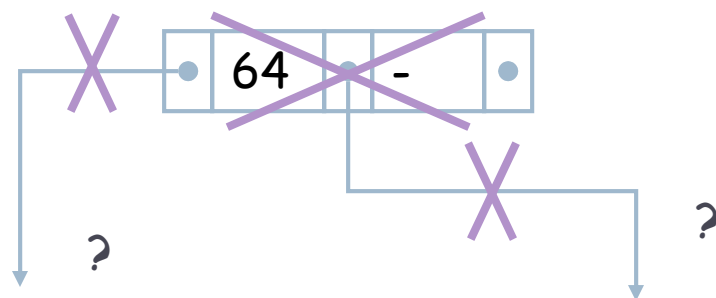
Brisanje zapisa s
ključem 64



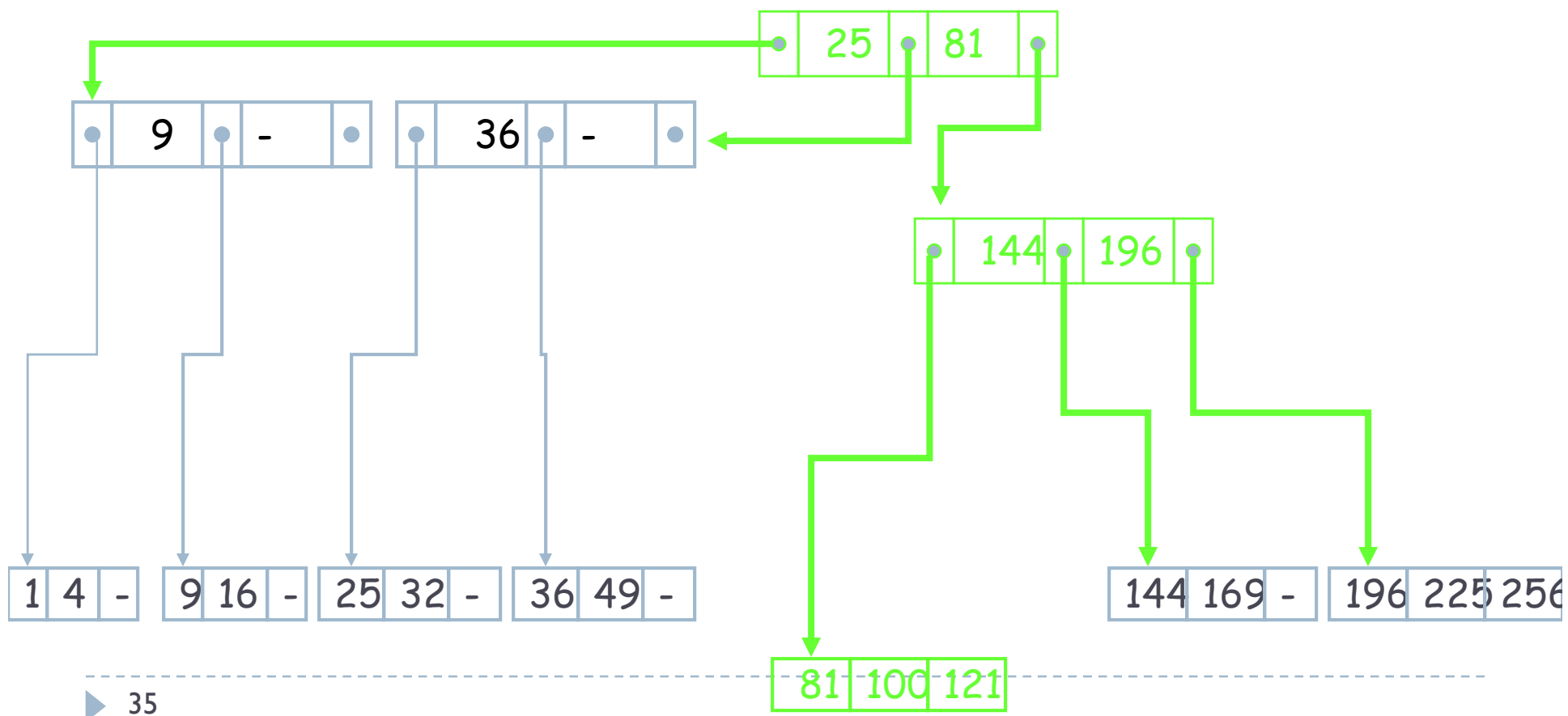
Brisanje zapisa s
ključem 64



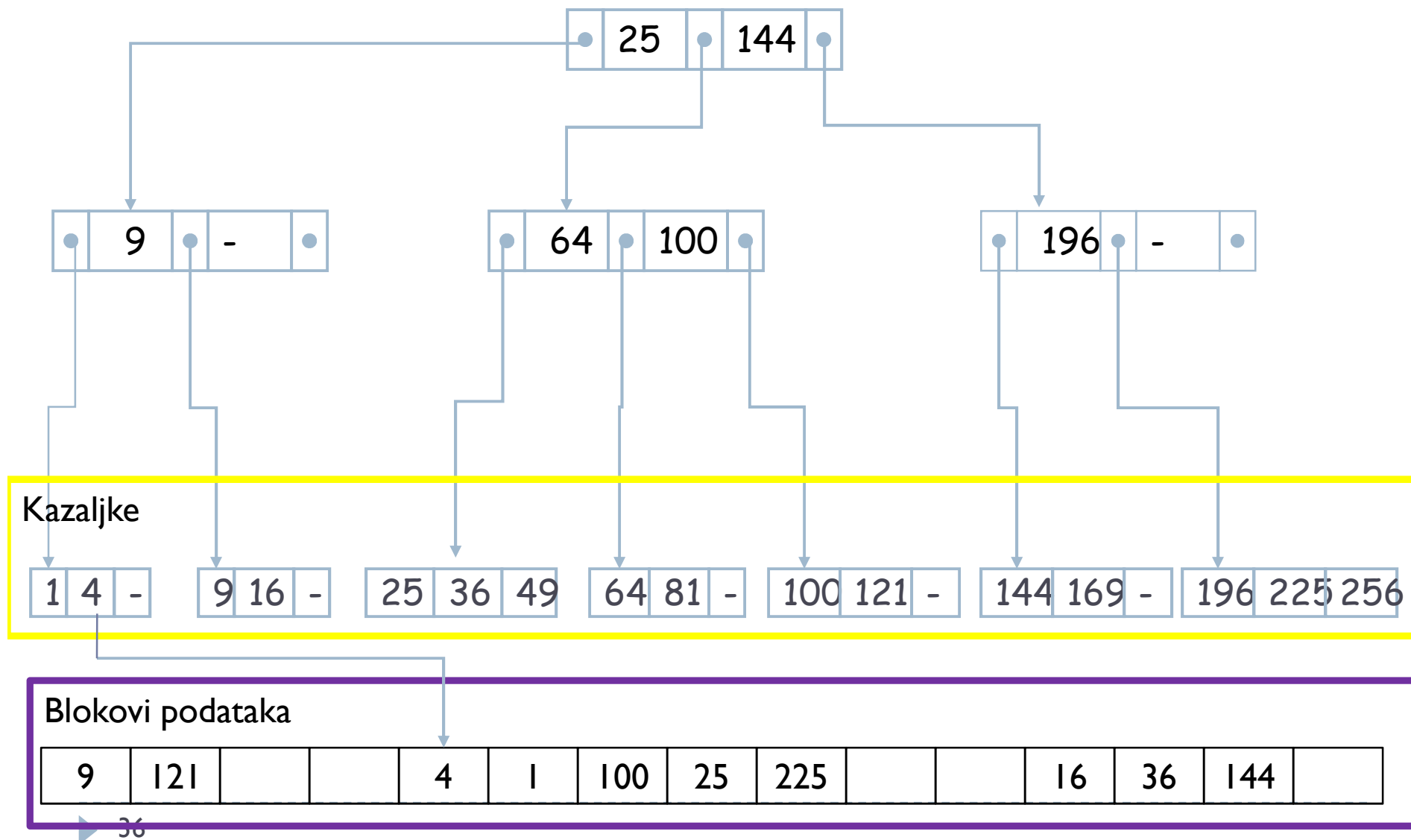
Brisanje zapisa s
ključem 64



Brisanje zapisa s
ključem 64



Najčeće se u listovima stabla NE nalaze podaci nego kazaljke na zapise.
 Time se omogućava da podaci ne moraju biti poredani (uporediti s indeksno
 Sekvencijalnom i indeksnom organizacijom sa direktnim pristupom)



Izračunavanje broja inspekcija potrebnih da bi se pristupilo zapisu s ključem k

- ▶ broj zapisa u indeksnom bloku = $2*d - 1$
- ▶ broj zapisa u podatkovnom bloku = $2*e - 1$
- ▶ broj zapisa u datoteci = n
 - ▶ popunjenost bloka u najgorem slučaju je d (odnosno e)
 - ▶ postoji najviše n / e listova
 - ▶ na svaki list pokazuje kazaljka s prethodnog nivoa - to znači da na prethodnom nivou ima najviše $(n / e) / d$ blokova, tj. $n / (d e)$ blokova
 - ▶ na sljedećem višem nivou ima najviše $n / (d^2 e)$ blokova
 - ▶ na i -tom nivou (gledano odozdo) ima $n / (d^{i-1} e)$ blokova
 - ▶ budući da broj blokova na svakom nivou mora biti veći ili jednak 1, vrijedi nejednačina $n / (d^{i-1} e) \geq 1$
 - ▶ odnosno $i \leq 1 + \log_d(n / e)$
- ▶ Primjer:
 - ▶ $n = 1\ 000\ 000$, $d = 50$, $e = 5$
 - ▶ Broj inspekcija u najgorem slučaju ≤ 5

