Indeksiranje

B+ indeks.

Uvod v indeksiranje (1)

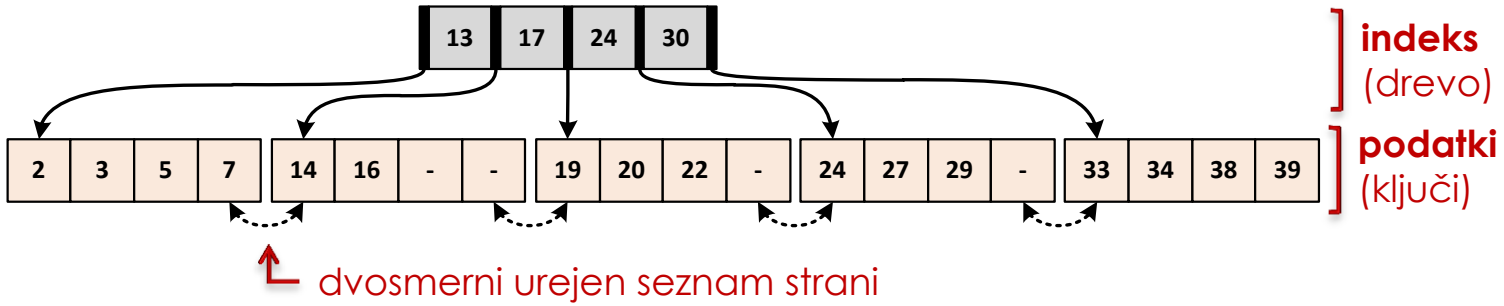
- Namen indeksiranja je **povečanje učinkovitosti sistema**.
  - Ponavadi želimo predvsem izboljšati učinkovitost (hitrost) poizvedb.
- Indeksi so **kazalci** (bližnjice) **na zapise**, da jih lahko hitreje najdemo in nam ni potrebno preiskati celotnih podatkov.

Uvod v indeksiranje (2)

- Pogledali si bomo naslednje metode indeksiranja, ki jih najpogosteje najdemo v podatkovnih bazah:
  - **B+** indeks (najbolj pogosto uporabljan v podatkovnih bazah),
  - **ISAM** indeks in
  - **bitni** indeks.

B+ indeks (1)

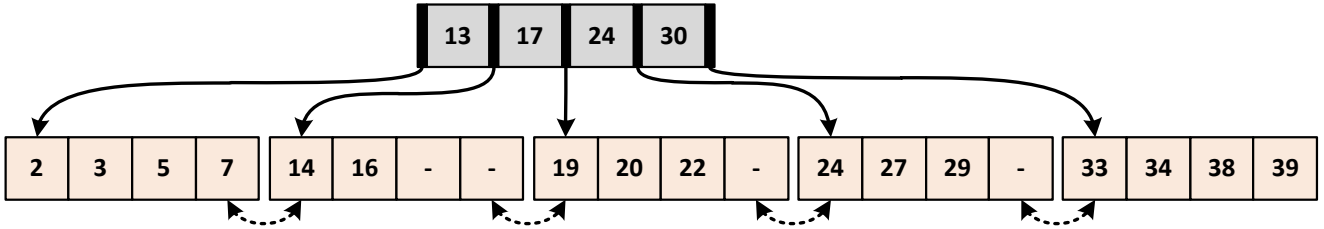
- **Dinamičen** = struktura se prilagaja spremembam v osnovni datoteki.
- **Uravnoteženo drevo**, kjer vozlišča usmerjajo iskanje, podatki (ključi) so shranjeni v listih.



- Operaciji vstavljanja in brisanja ohranjata drevo uravnoteženo.

B+ indeks (2)

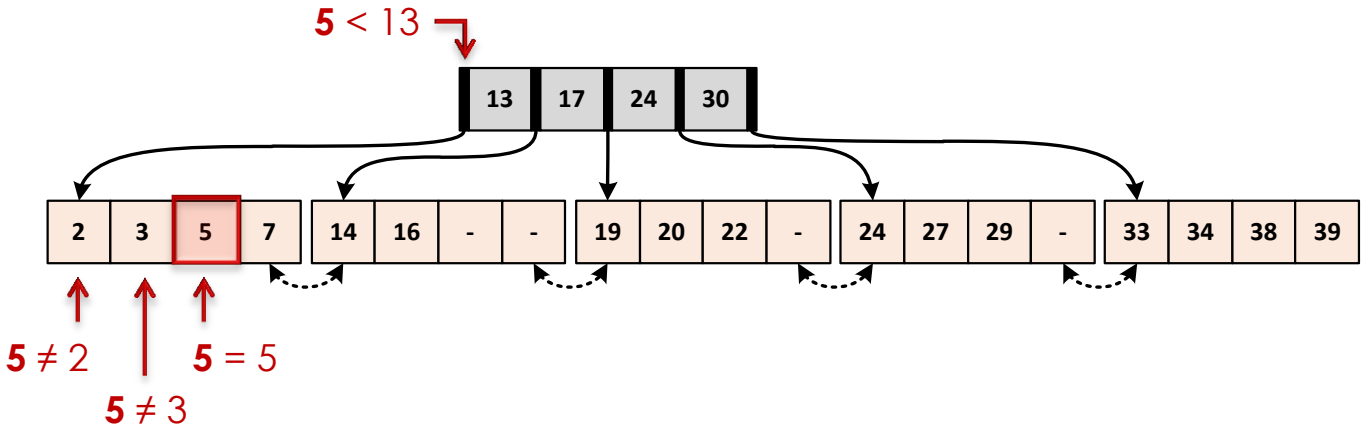
- Pogoj vozlišča: **d ≤ m ≤ 2d** (vsaj 50% zasedenost)
  - **m** … število zapisov (zasedenost) vozlišča
  - **d** … red drevesa (2d = kapaciteta vozlišča)
  - Koren je izjema: **1 ≤ m ≤ 2d**



Red drevesa **d=2**.
Zasedenost vozlišča (indeksa) je **vsaj 2** in **največ 4** zapisi.

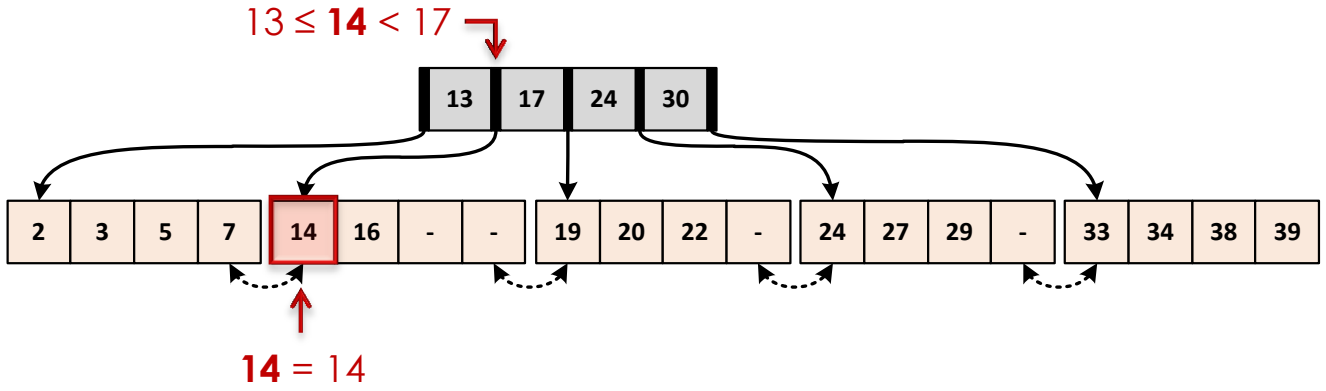
B+ indeks – primer iskanja (1)

- Poišči zapis s ključem ***5***.



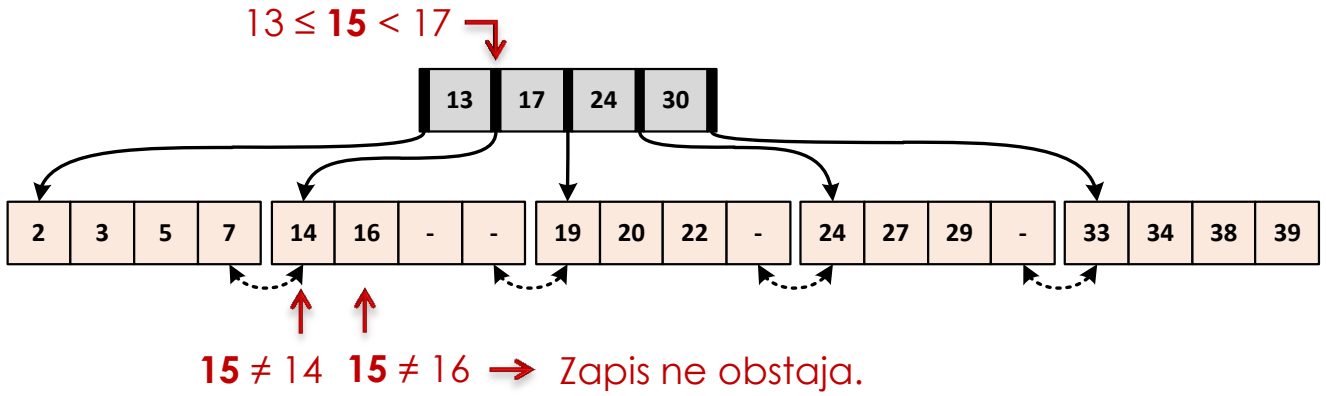
B+ indeks – primer iskanja (2)

- Poišči zapis s ključem **14**.



B+ indeks – primer iskanja (3)

- Poišči zapis s ključem **15**.

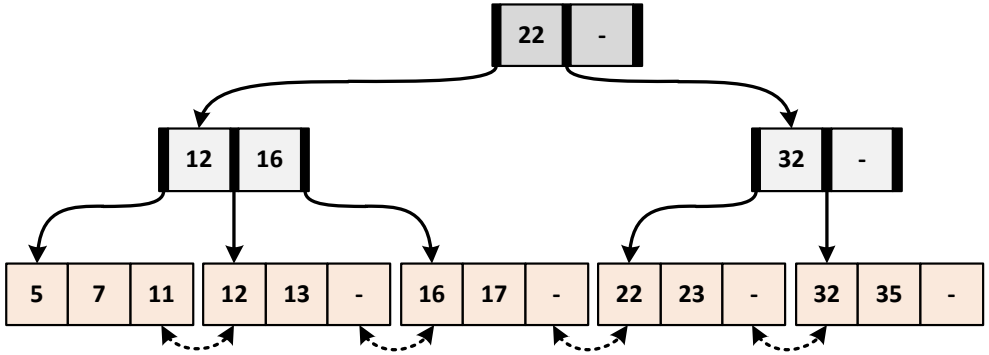


B+ indeks – dodajanje

- **Algoritem dodajanja zapisa k***
  - **POIŠČI** list, kamor k* spada
  - **ČE** je v listu prostor, k* dodamo
  - **SICER** list **razdelimo** na 2 dela:
    - d elementov gre v levi list
    - preostanek gre v desni list
  - Po potrebi popravimo delitveni ključ navzgor po drevesu indeks.
- Poznamo 2 različici: **brez** in **z redistribucijo**
  - Če ni opredeljeno, se uporabi brez redistribucije.

B+ indeks primer 1 (1)

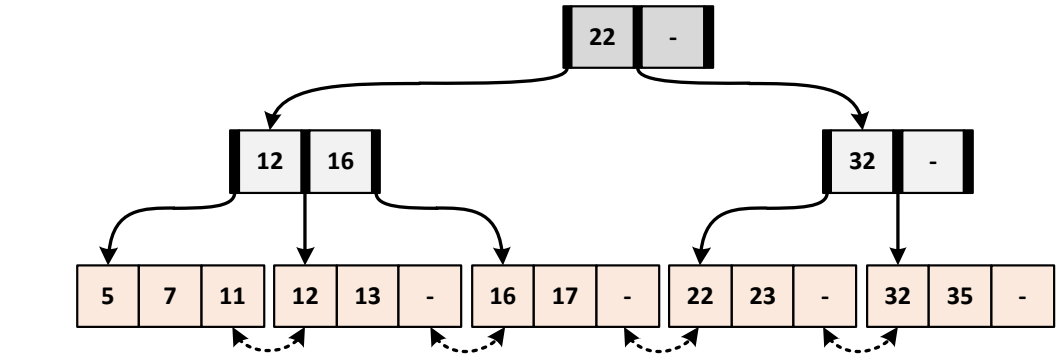
- Nad osnovno datoteko je zgrajen B+ indeks, ki ga prikazuje slika.



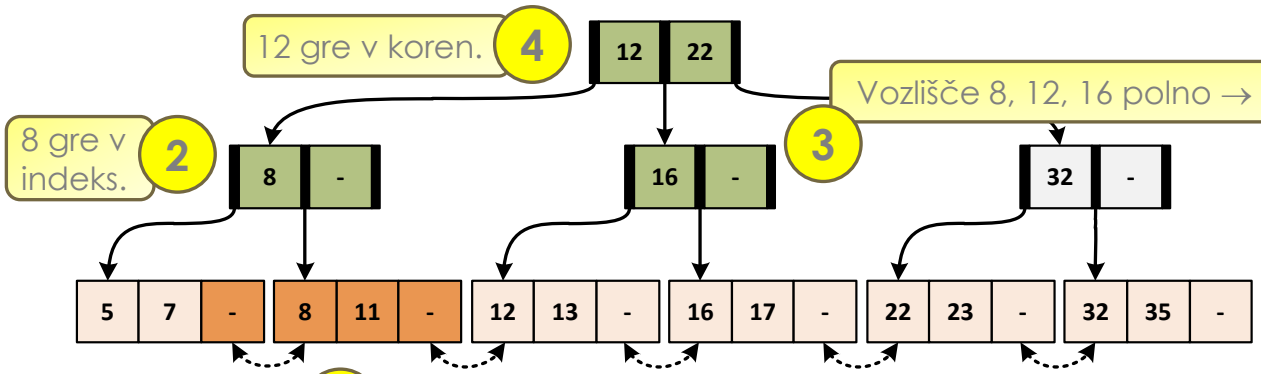
- **Dodaj** zapise s ključi **8**, **24** in **25** enkrat **brez redistribucije** in drugič **z redistribucijo**!

B+ indeks primer 1.a (2)

- **Dodaj** zapis s ključem **8 brez redistribucije**.



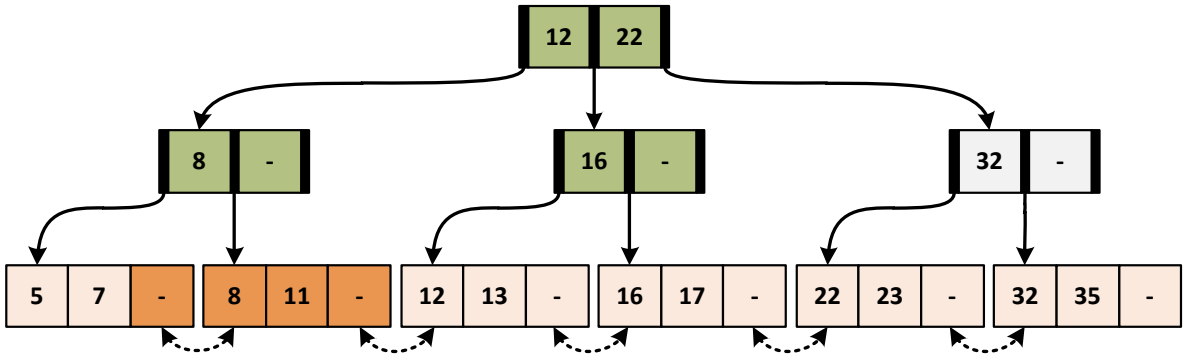
PRED operacijo



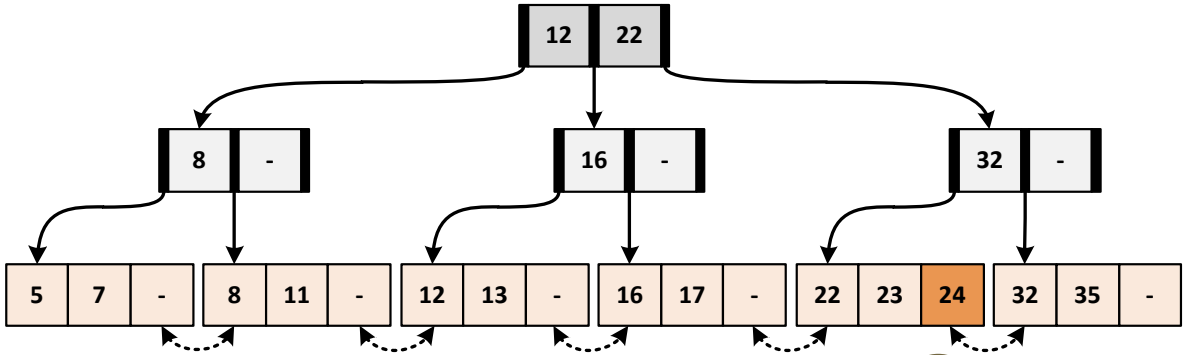
PO operaciji

B+ indeks primer 1.a (3)

- **Dodaj** zapis s ključem **24 brez redistribucije**.



PRED operacijo

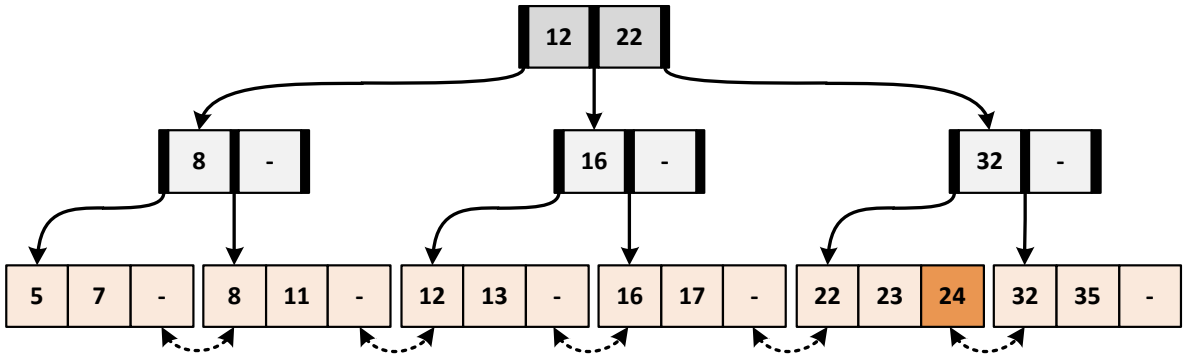


PO operaciji

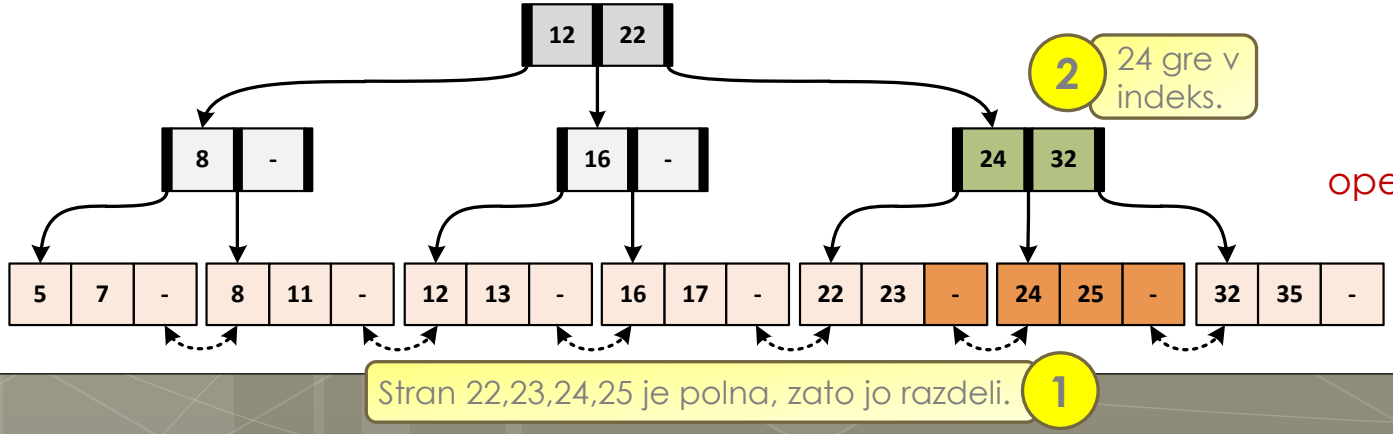
1 Dodaj na prosto mesto.

B+ indeks primer 1.a (4)

- **Dodaj** zapis s ključem **25 brez redistribucije**.



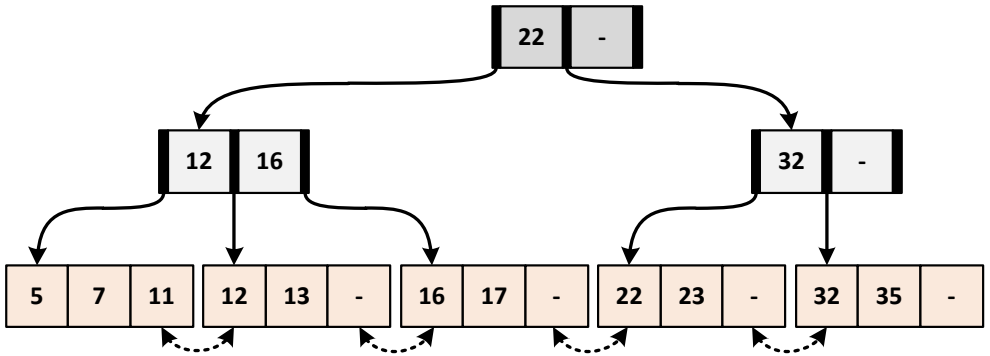
PRED operacijo



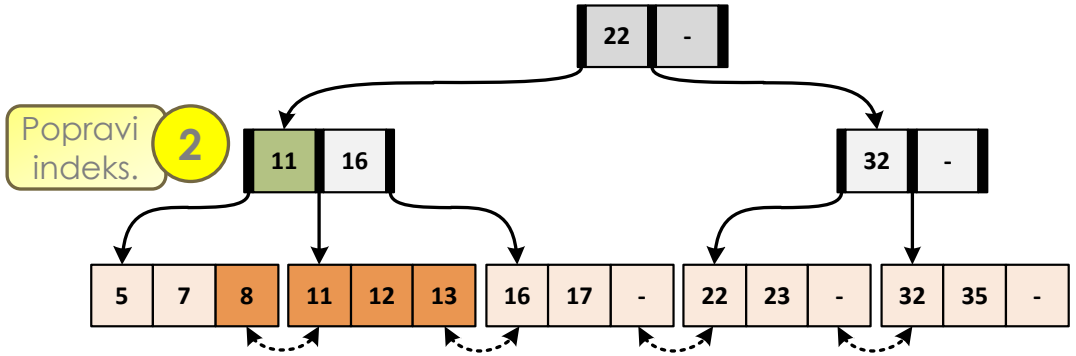
PO operaciji

B+ indeks primer 1.b (5)

- **Dodaj** zapis s ključem **8 z redistribucijo**.



PRED operacijo

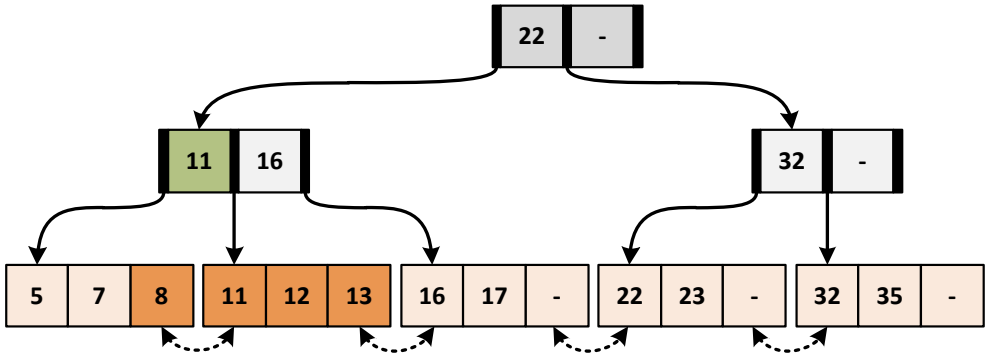


PO operaciji

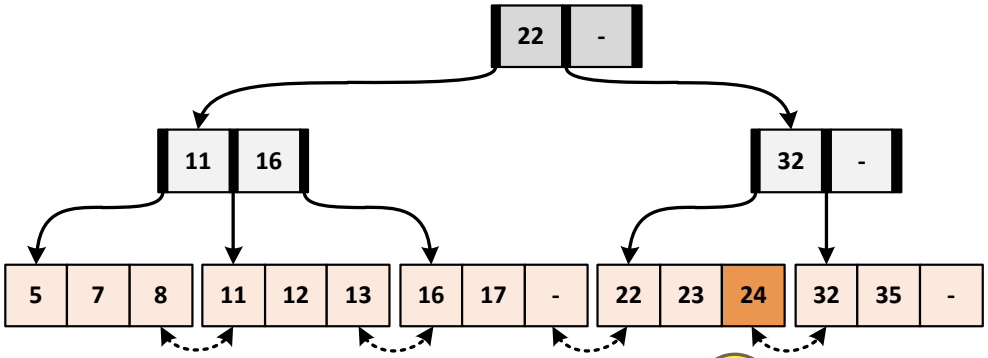
1 Stran je polna, a ima naslednja stran prosto mesto, zato uporabimo redistribucijo.

B+ indeks primer 1.b (6)

- **Dodaj** zapis s ključem **24 z redistribucijo**.



PRED operacijo

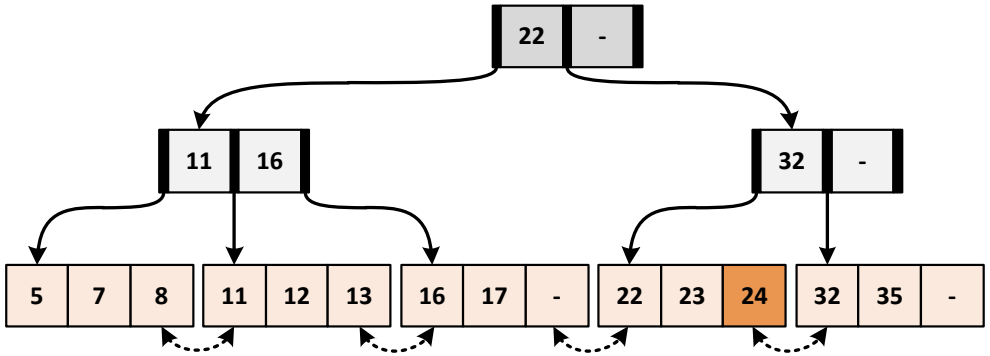


PO operaciji

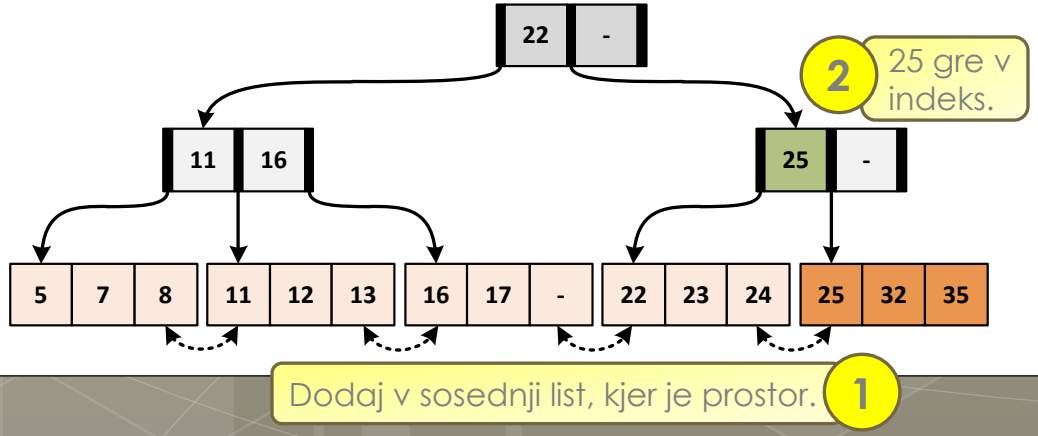
Dodaj na prosto mesto. 1

B+ indeks primer 1.b (6)

- **Dodaj** zapis s ključem **25 z redistribucijo**.



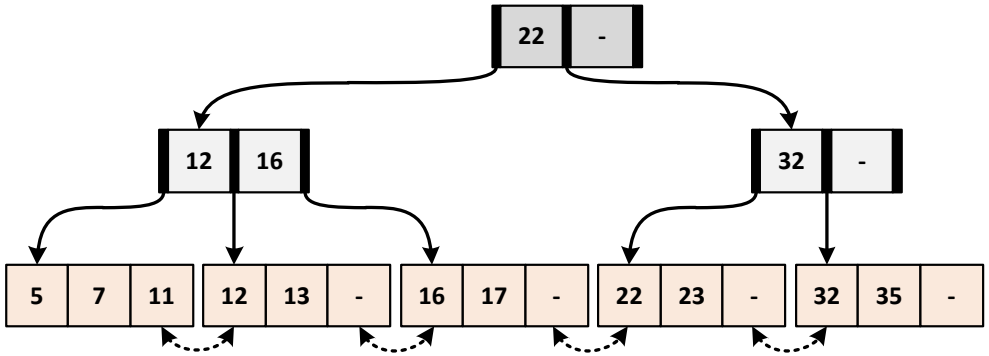
PRED operacijo



PO operaciji

B+ indeks primer 2 (7)

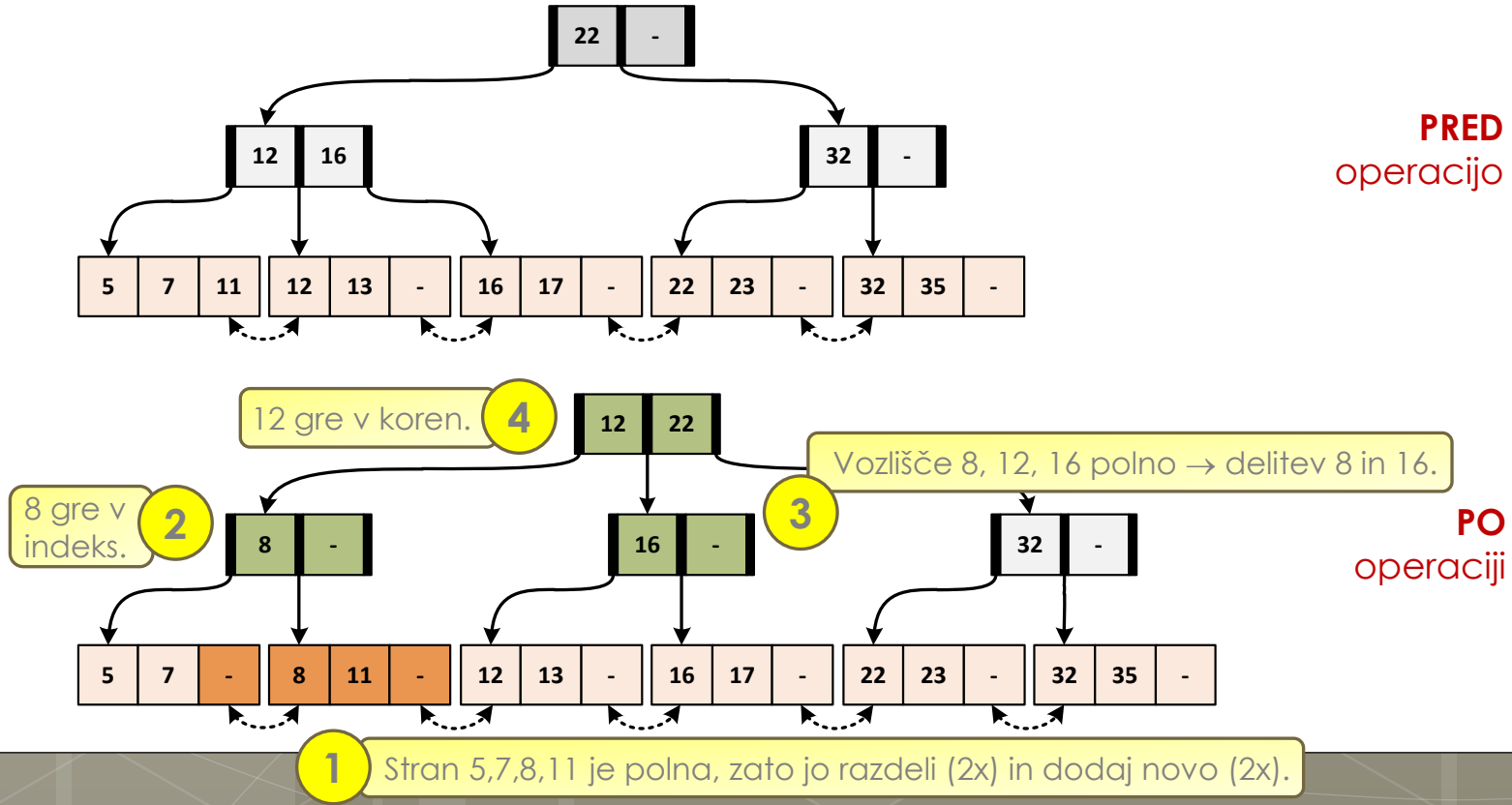
- Nad osnovno datoteko je zgrajen B+ indeks, ki ga prikazuje slika.



- Najprej **dodaj** zapis s ključem **8**, nato pa **izbriši** zapis s ključem **32**.

B+ indeks primer 2 (8)

- **Dodaj** zapis s ključem **8**.

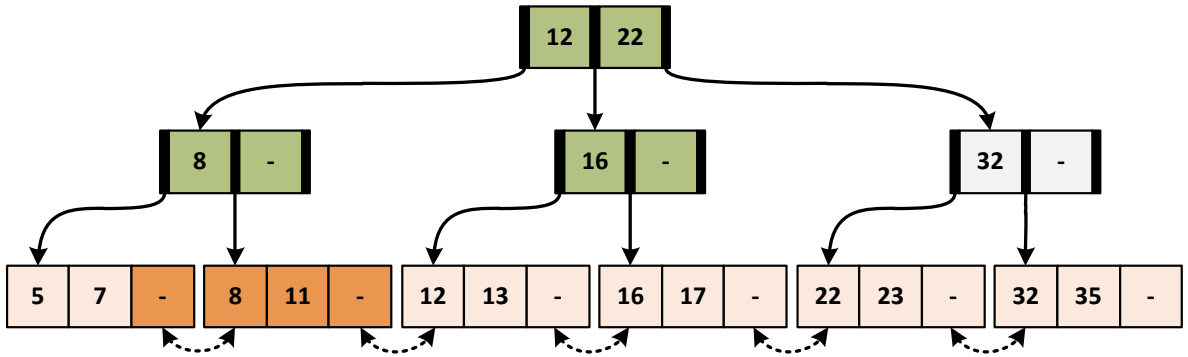


PRED operacijo

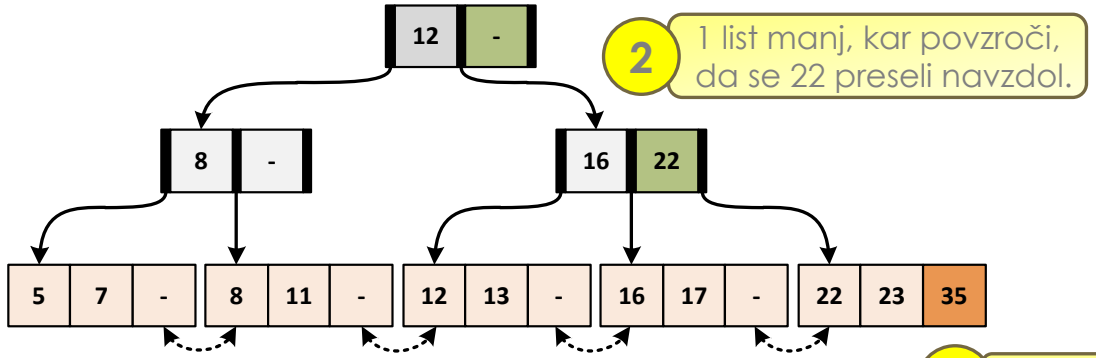
PO operaciji

B+ indeks primer 2 (9)

- **Izbriši** zapis s ključem **32**.



PRED operacijo



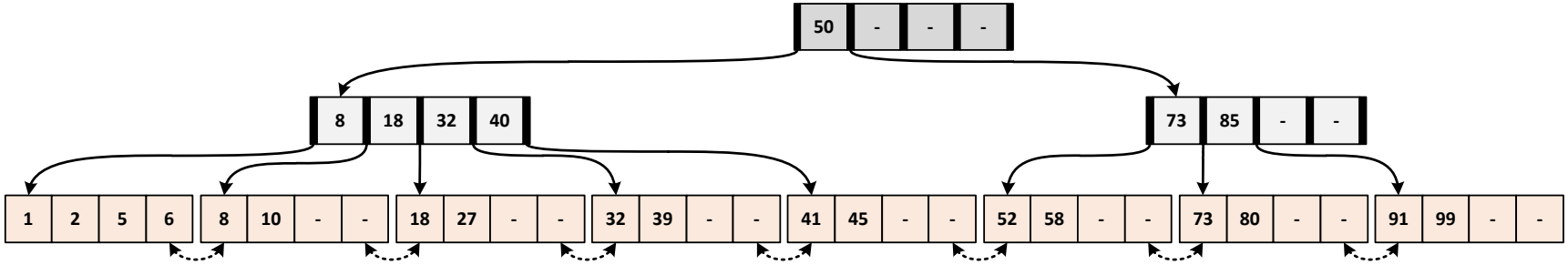
2 1 list manj, kar povzroči, da se 22 preseli navzdol.

PO operaciji

1 Združi sosednji strani.

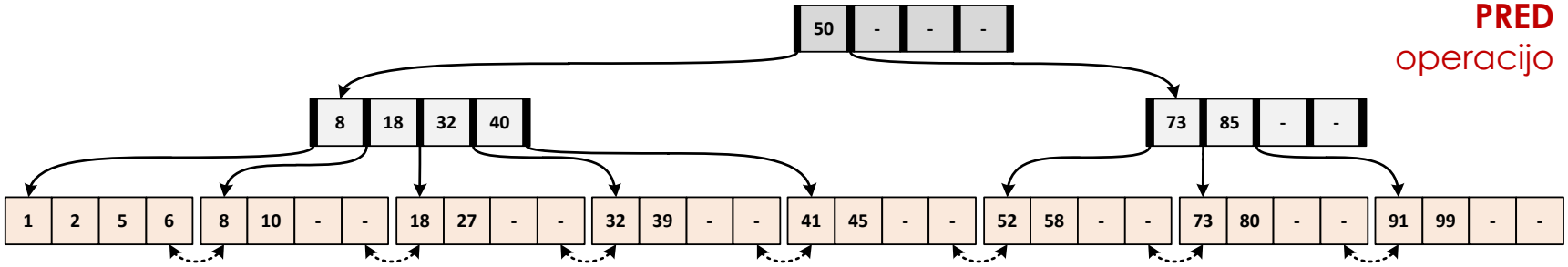
B+ indeks naloga (1)

- Nad osnovno datoteko je zgrajen B+ indeks, ki ga prikazuje slika.
- V nadaljevanju je potrebno izvesti določene operacije in prikazati rezultat v obliki drevesa.

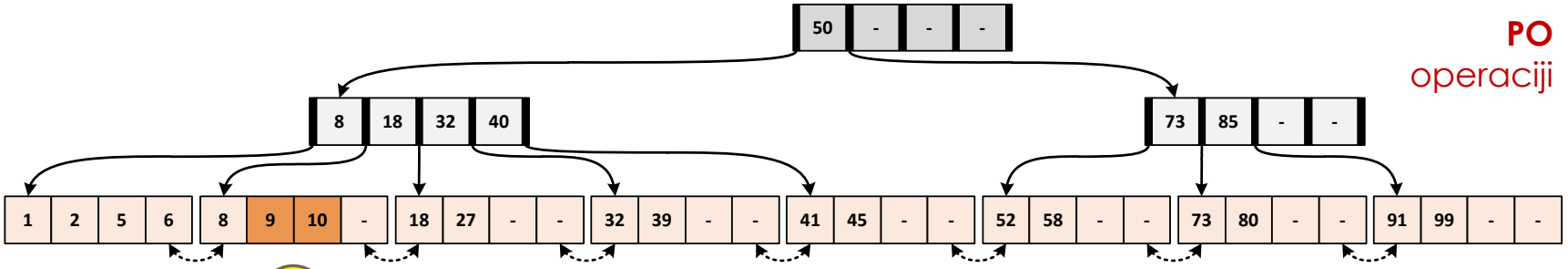


B+ indeks naloga 1 (2)

- **Dodaj** zapis s ključem **9**.



PRED operacijo

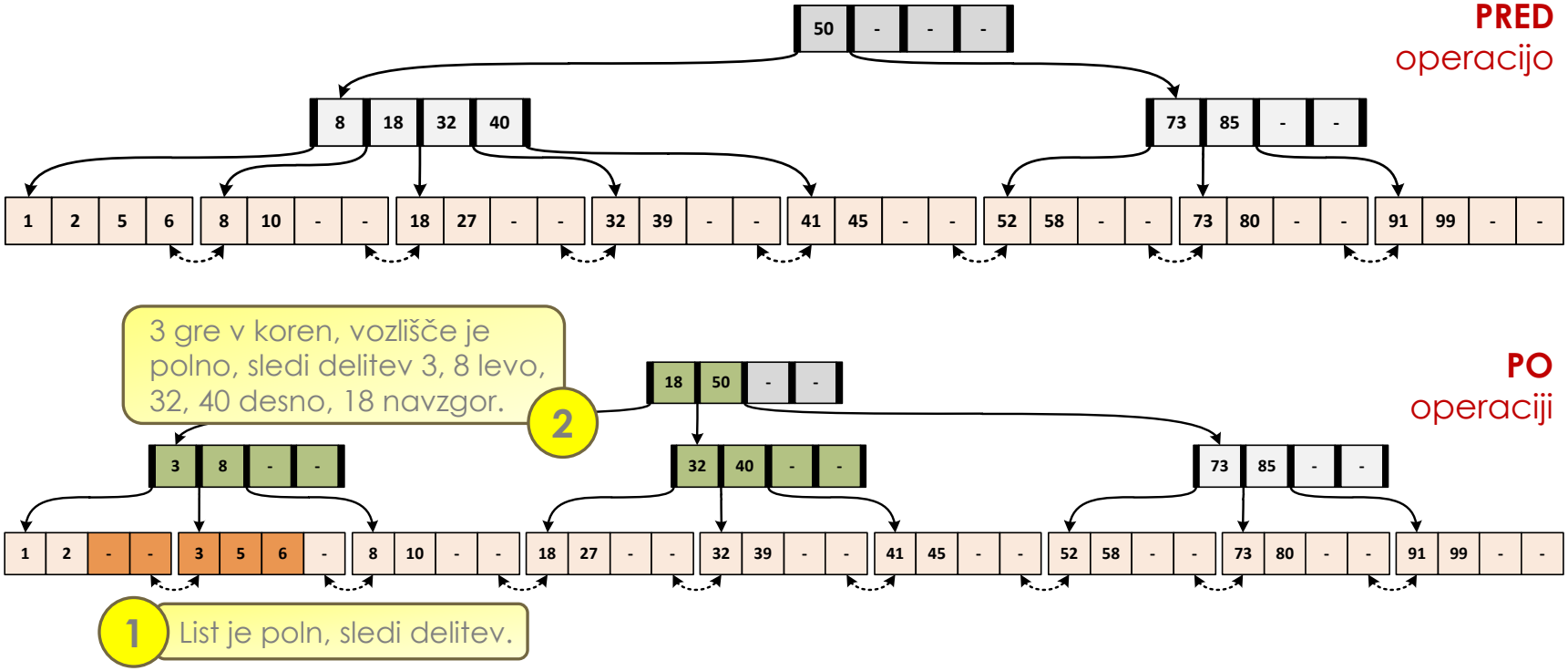


PO operaciji

1. Dodaj na prosto mesto.

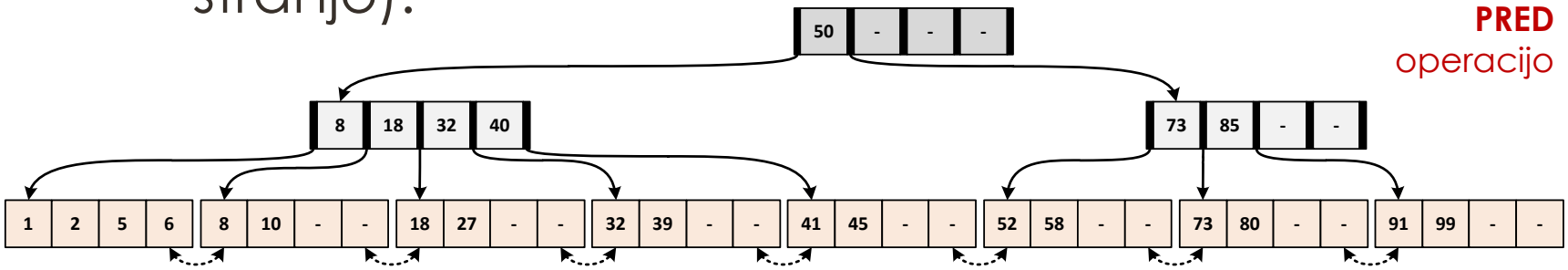
B+ indeks naloga 2 (3)

- **Dodaj** zapis s ključem **3**.

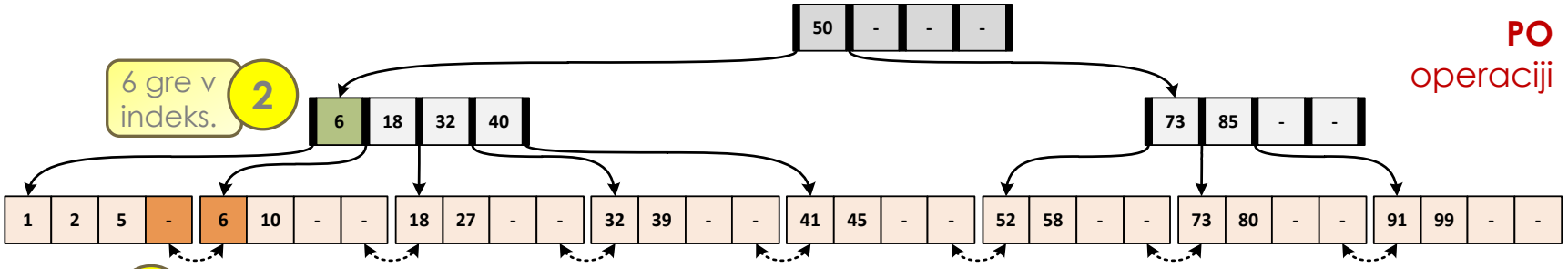


B+ indeks naloga 3 (4)

- **Izbriši** zapis s ključem **8** (redistribucija z levo stranjo).



PRED operacijo



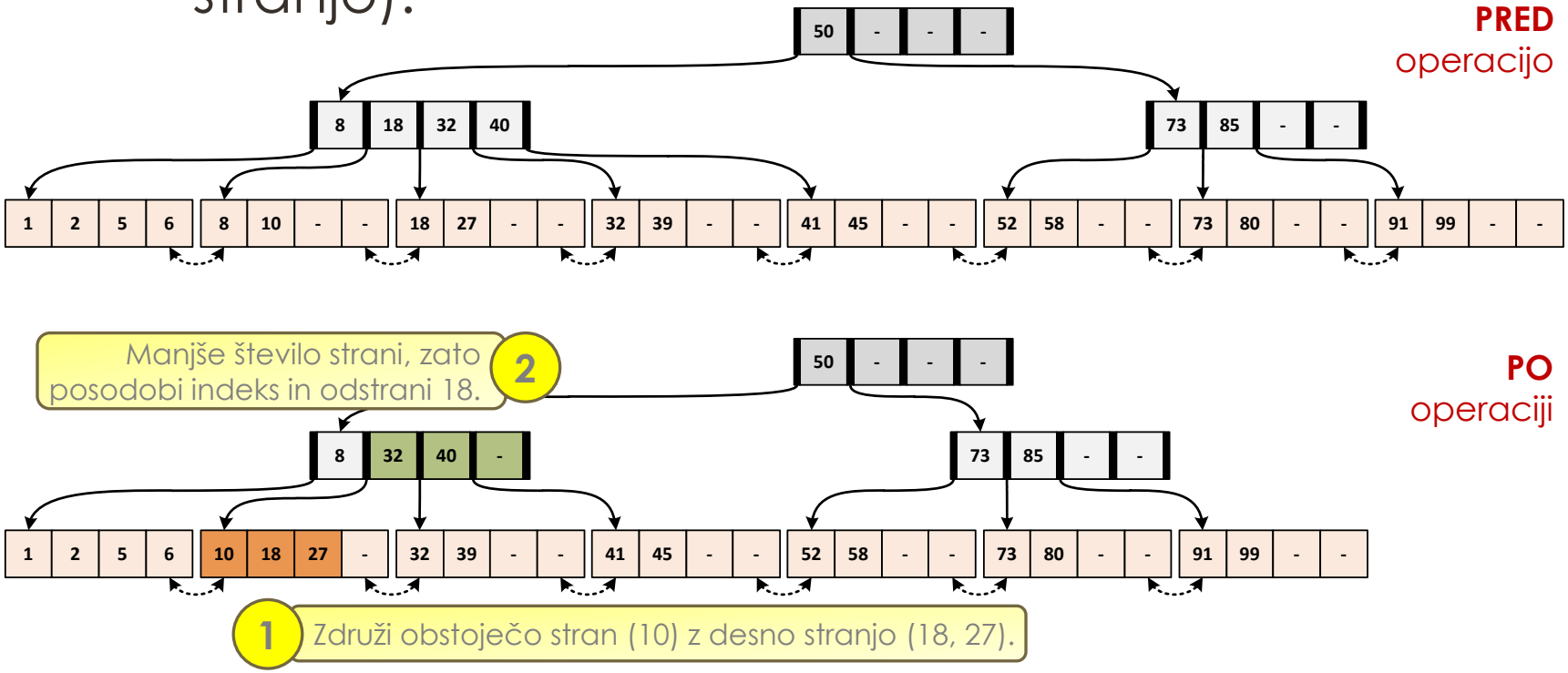
PO operaciji

6 gre v indeks. 2

1 Iz levega vozlišča prenesi 1 zapis, da je izpolnjen pogoj za obstoj vozlišča.

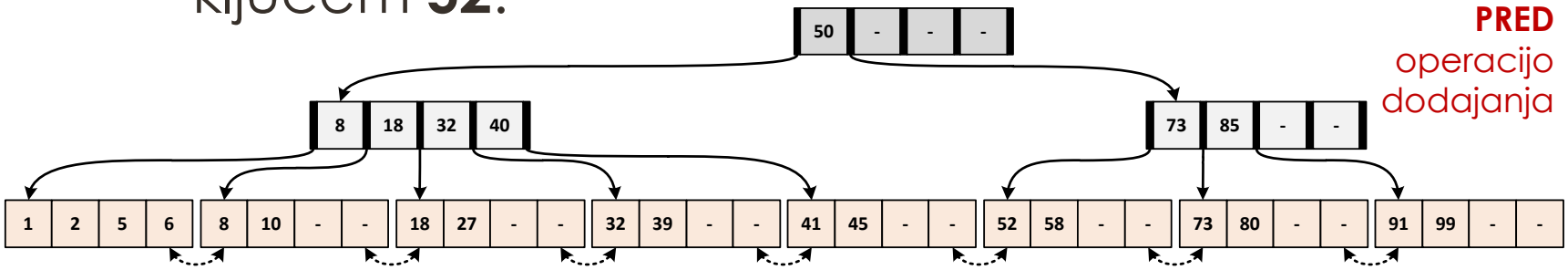
B+ indeks naloga 4 (5)

- **Izbriši** zapis s ključem **8** (redistribucija z desno stranjo).

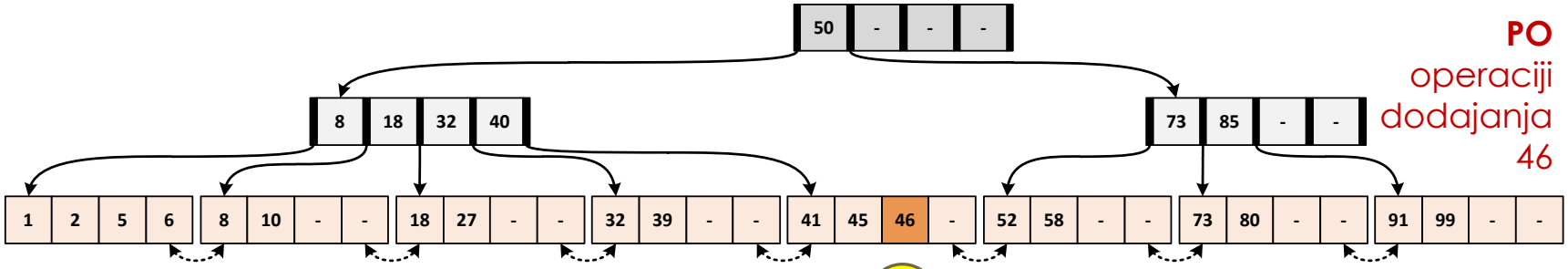


B+ indeks naloga 5 (6)

- **Dodaj** zapis s ključem **46** in nato **izbriši** zapis s ključem **52**.



PRED operacijo dodajanja

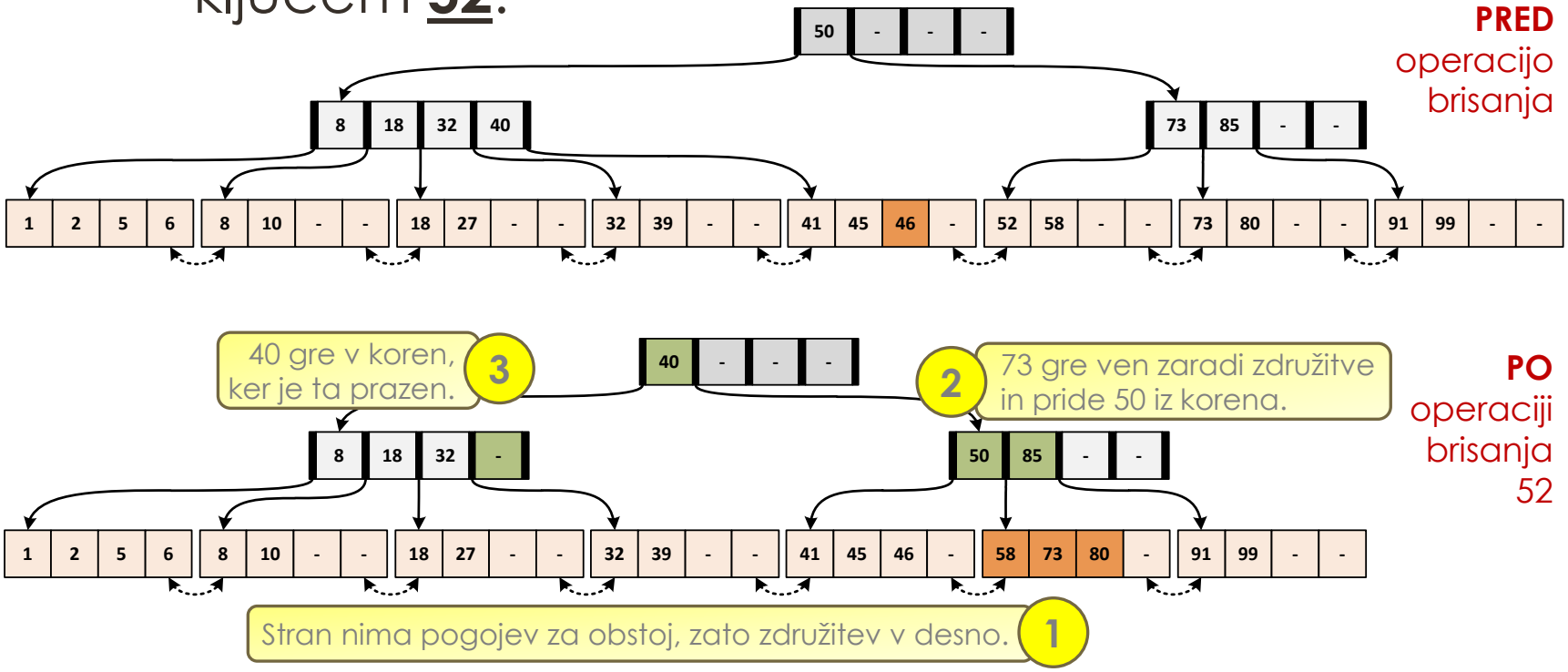


PO operaciji dodajanja 46

Dodaj na prosto mesto. 1

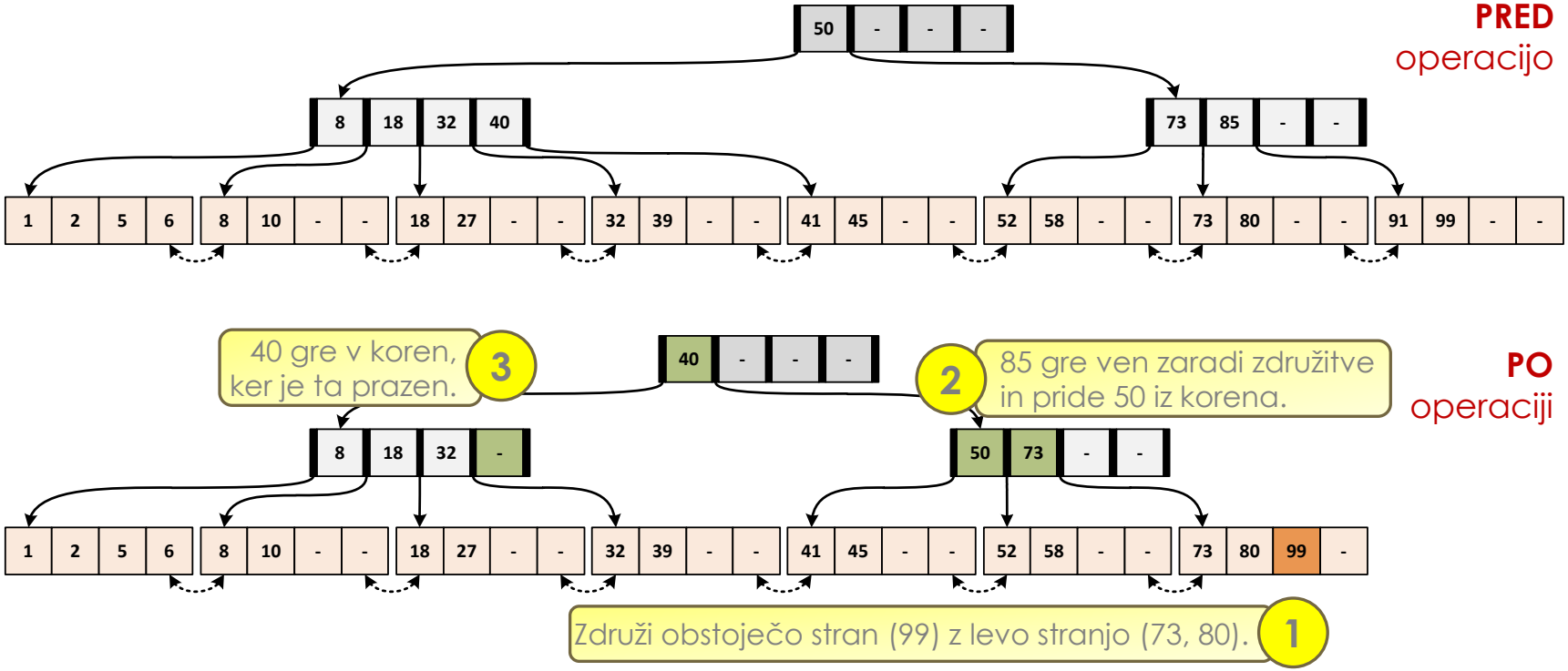
B+ indeks naloga 5 (7)

- **Dodaj** zapis s ključem **46** in nato **izbriši** zapis s ključem **52**.



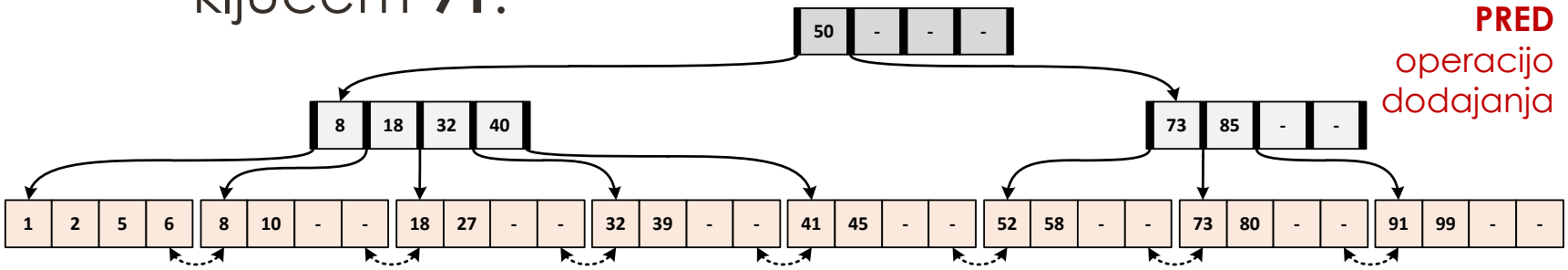
B+ indeks naloga 6 (8)

- **Izbriši** zapis s ključem **91**.

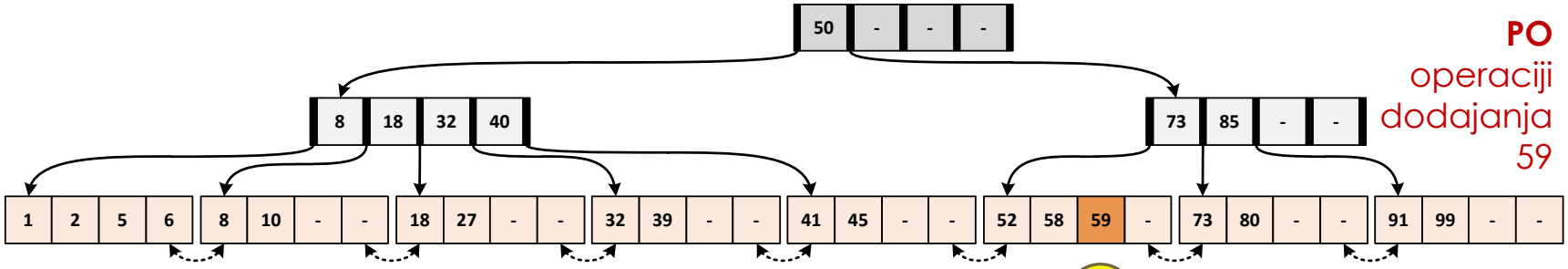


B+ indeks naloga 7 (9)

- **Dodaj** zapis s ključem **59** in nato **izbriši** zapis s ključem **91**.



PRED operacijo dodajanja

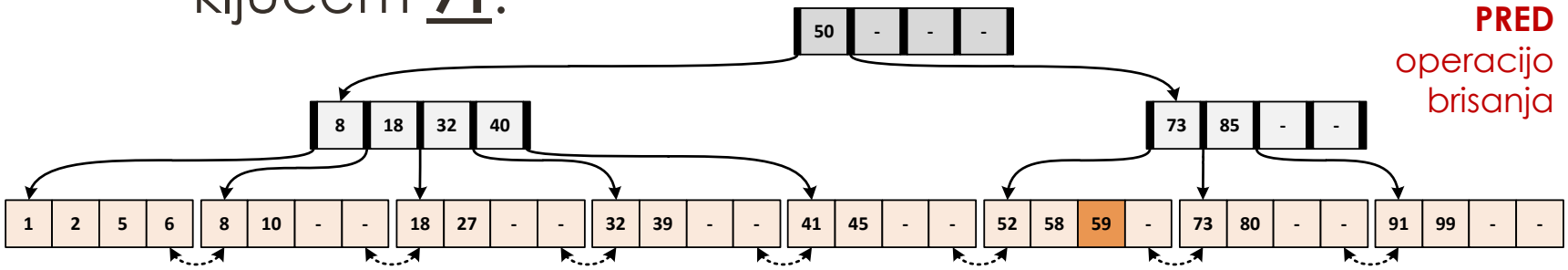


PO operaciji dodajanja 59

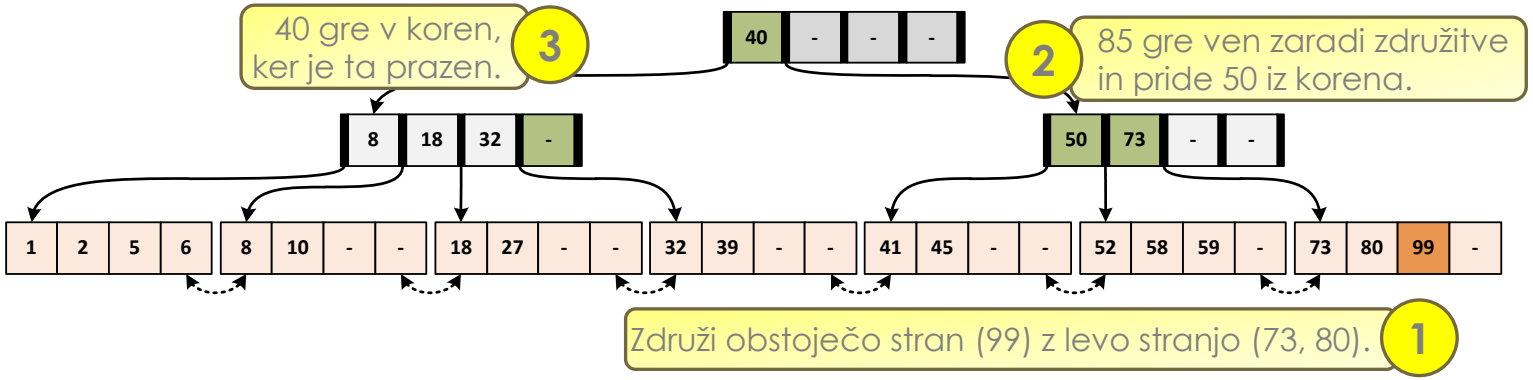
Dodaj na prosto mesto. 1

B+ indeks naloga 7 (10)

- **Dodaj** zapis s ključem **59** in nato **izbriši** zapis s ključem **91**.



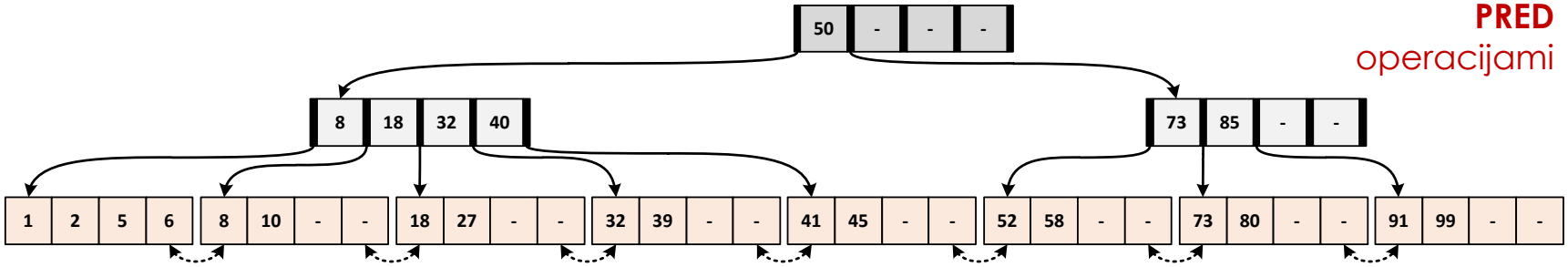
PRED operacijo brisanja



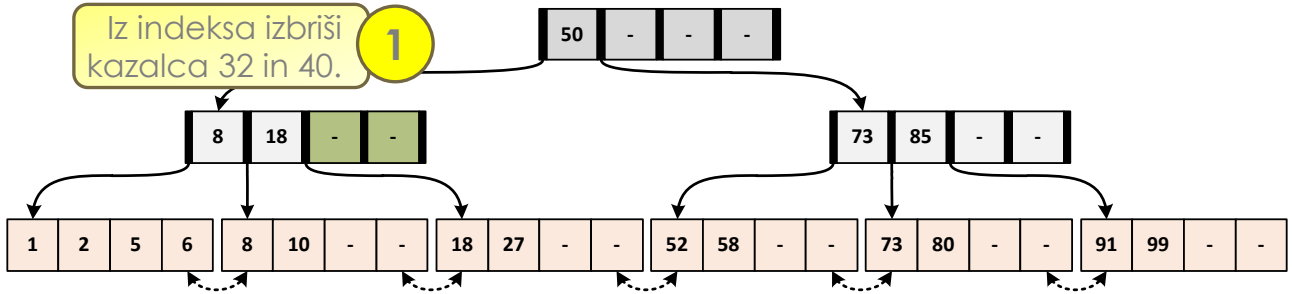
PO operaciji brisanja 91

B+ indeks naloga 8 (11)

- **Izbriši** zapise s ključi **32**, **39**, **41**, **45** in **73**.



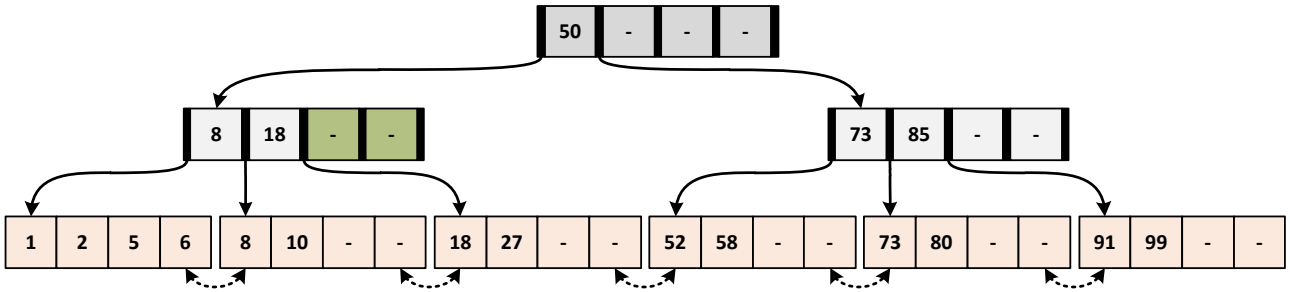
PRED operacijami



PO operacijah brisanja 32, 39, 41 in 45

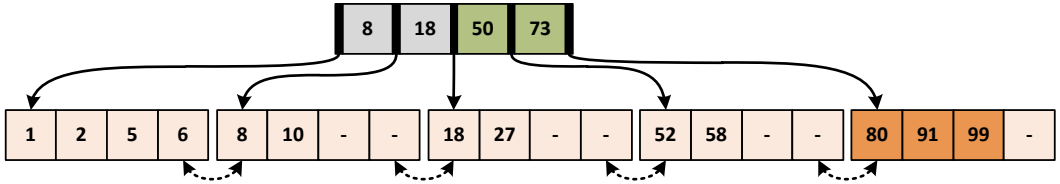
B+ indeks naloga 8 (12)

- **Izbriši** zapise s ključi **32**, **39**, **41**, **45** in **73**.



PRED zadnjo operacijo brisanja

2 Odstrani indeks 85, nato povleči 50 iz korena in združi obe vozlišči.



Združi obstoječo stran (99) z levo stranjo (73, 80). 1

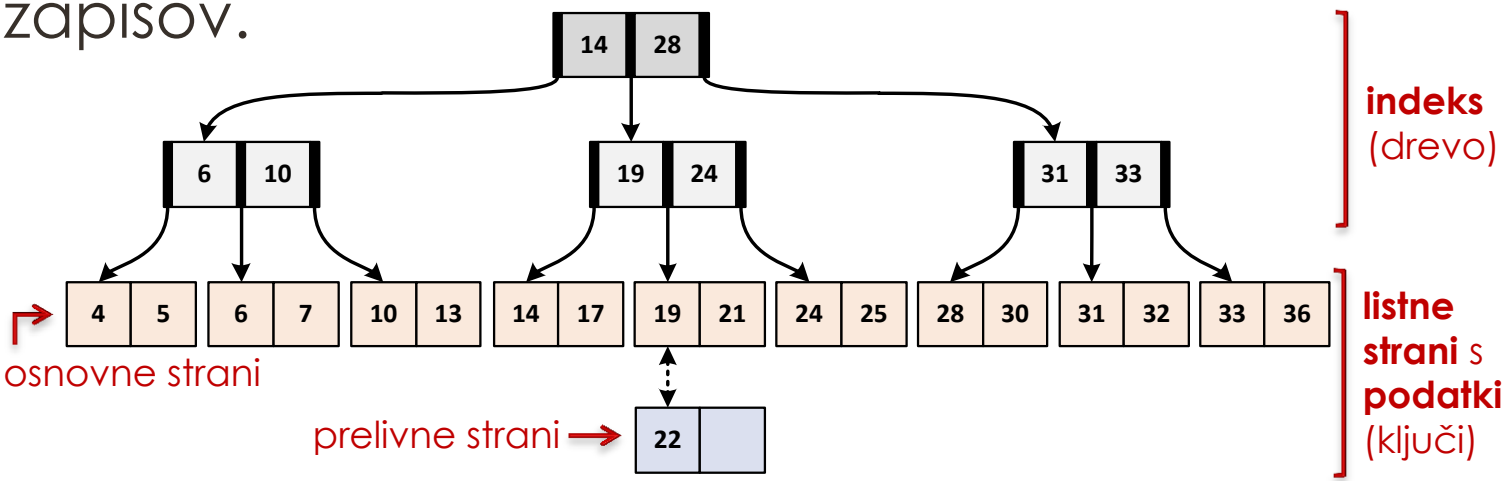
PO operaciji brisanja 73

Indeksiranje

ISAM indeks.

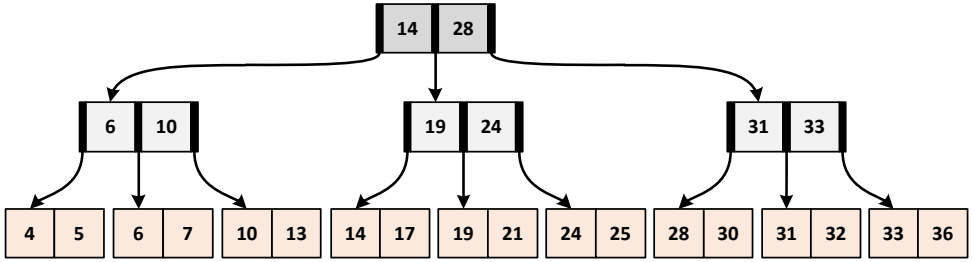
ISAM indeks (1)

- **Statičen** (z izjemo prelivnih strani) = potrebno ga je reorganizirati.
- Podatki v prelivnih straneh načeloma niso urejeni → zaradi učinkovitosti dodajanja zapisov.



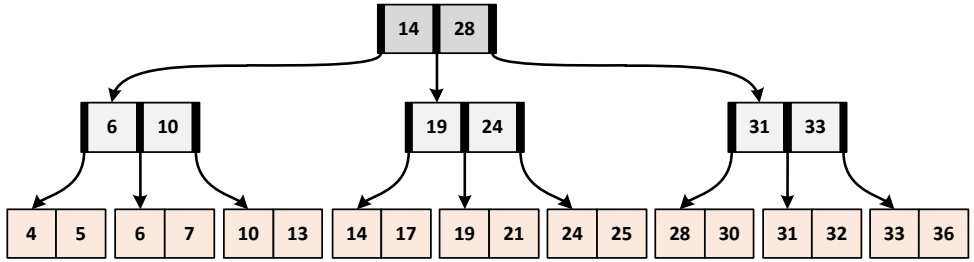
ISAM indeks naloga 1 (1)

- Nad osnovno datoteko je zgrajen ISAM indeks.
- V nadaljevanju je potrebno izvesti določene operacije in prikazati rezultat v obliki drevesa.

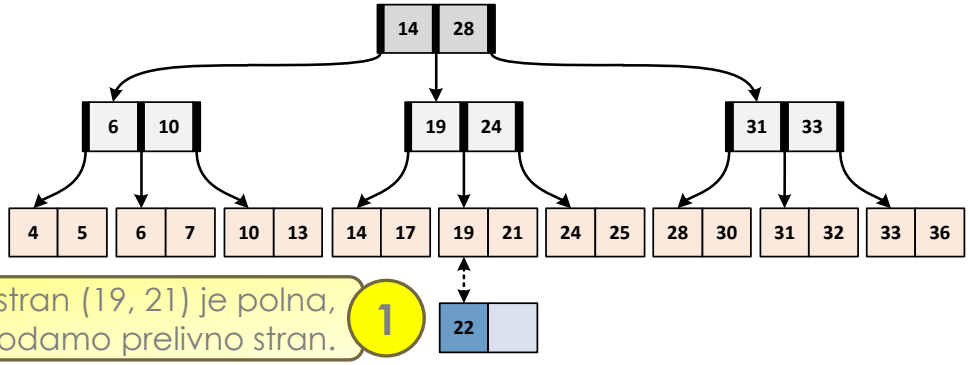


ISAM indeks naloga 1.a (2)

- **Dodaj** zapis s ključem **22** in **20**.



PRED operacijo

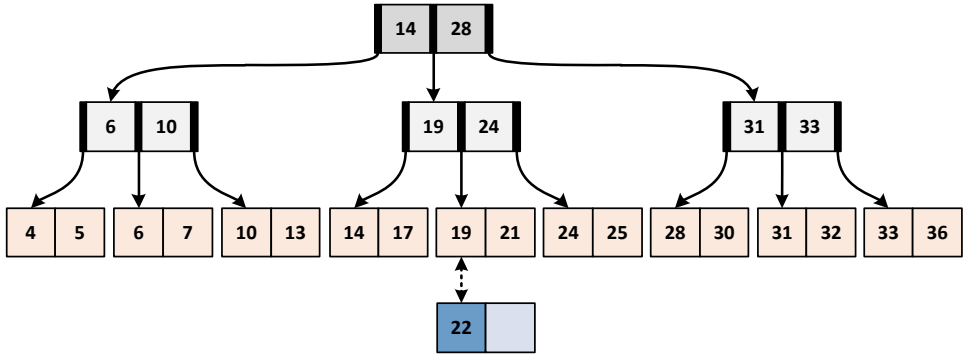


Osnovna stran (19, 21) je polna, zato dodamo prelivno stran. 1

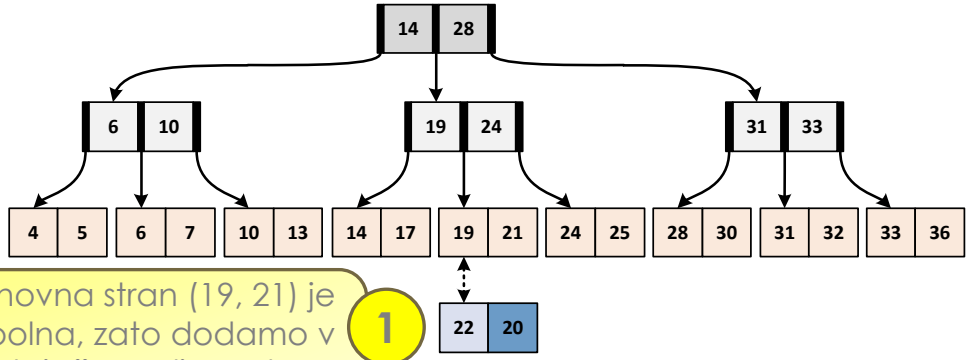
PO operaciji dodajanja 22

ISAM indeks naloga 1.a (3)

- **Dodaj** zapis s ključem **22** in **20**.



PRED operacijo dodajanja 20



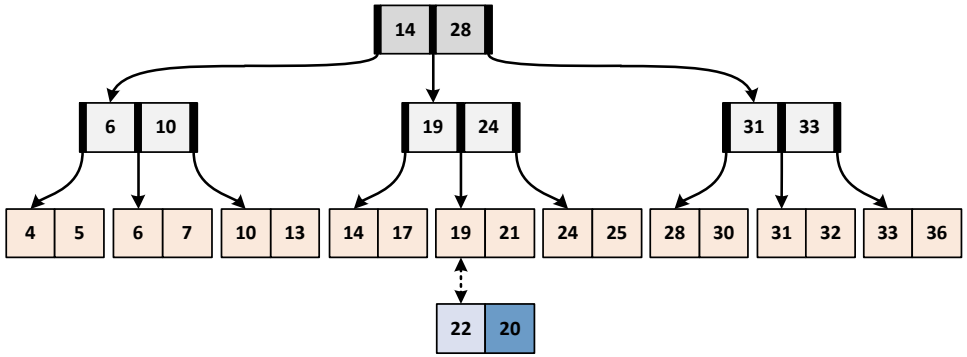
PO operaciji dodajanja 20

Osnovna stran (19, 21) je polna, zato dodamo v obstoječo prelivno stran, kjer vrednosti ne urejamo.

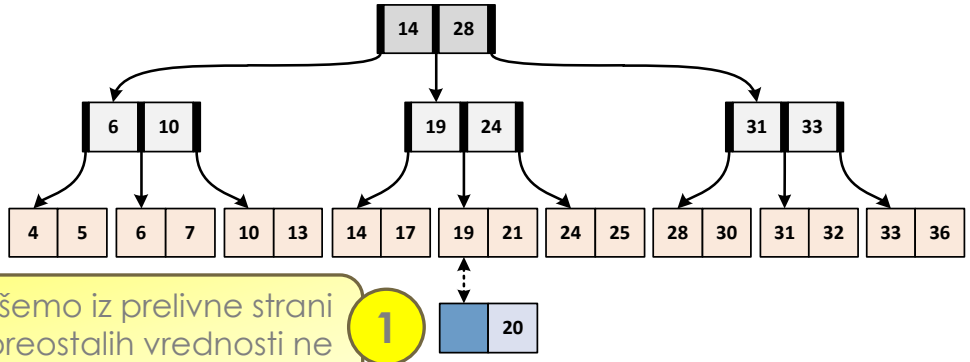
1

ISAM indeks naloga 1.b (4)

- **Izbriši** zapise s ključi **22**, **20**, **21**, **19** in **31**.



PRED operacijo



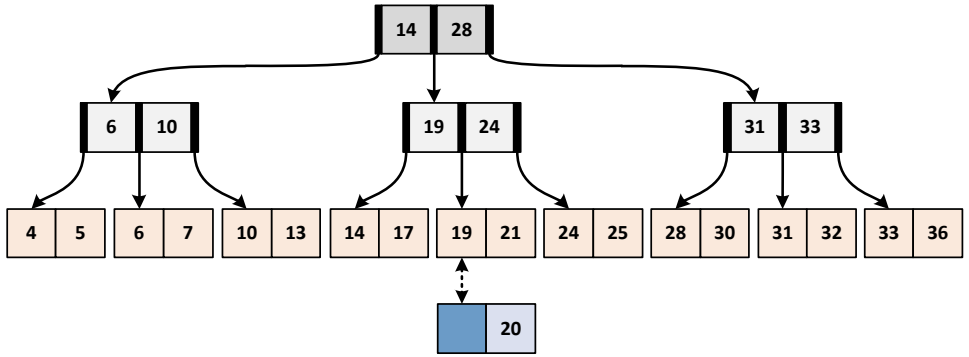
PO operaciji brisanja 22

Izbrišemo iz prelivne strani in preostalih vrednosti ne preurejamo.

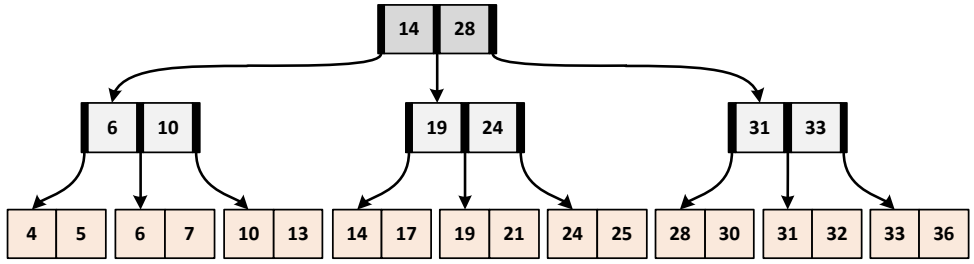
1

ISAM indeks naloga 1.b (5)

- **Izbriši** zapise s ključi **22**, **20**, **21**, **19** in **31**.



PRED operacijo



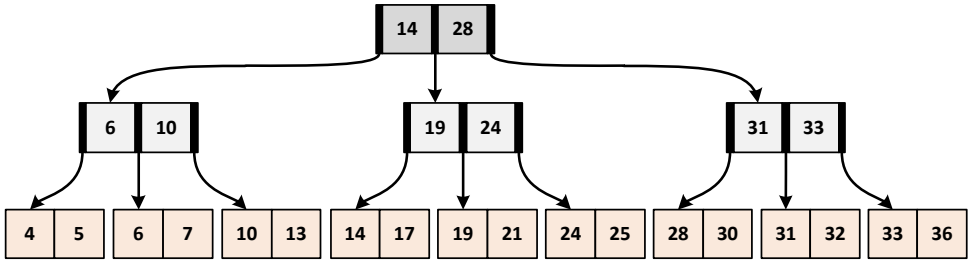
PO operaciji brisanja 20

Izbrisali smo zadnji element prelivne strani, zato izbrišemo tudi prelivno stran.

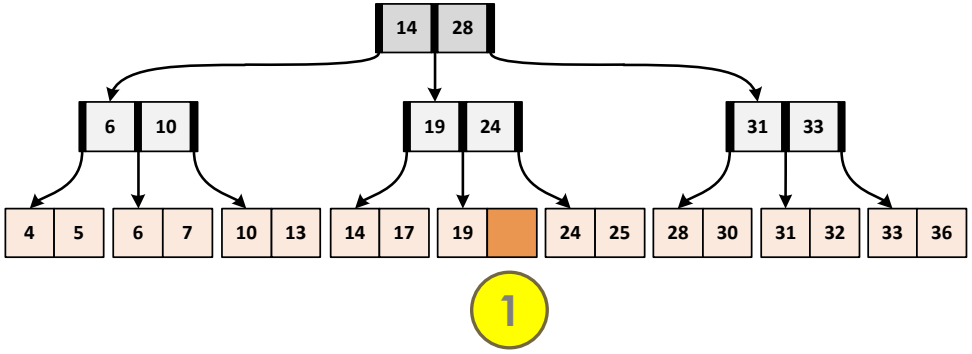
1

ISAM indeks naloga 1.b (6)

- **Izbriši** zapise s ključi **22**, **20**, **21**, **19** in **31**.



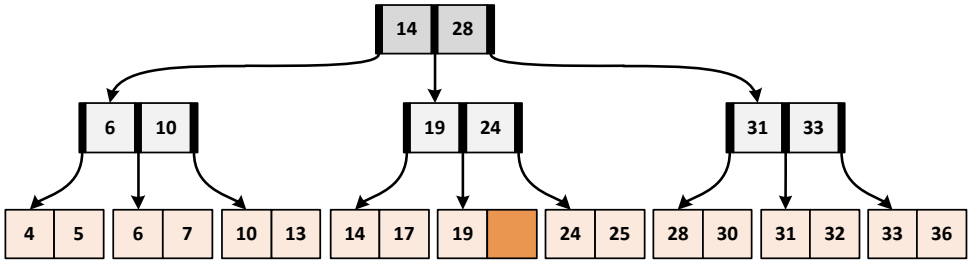
PRED operacijo



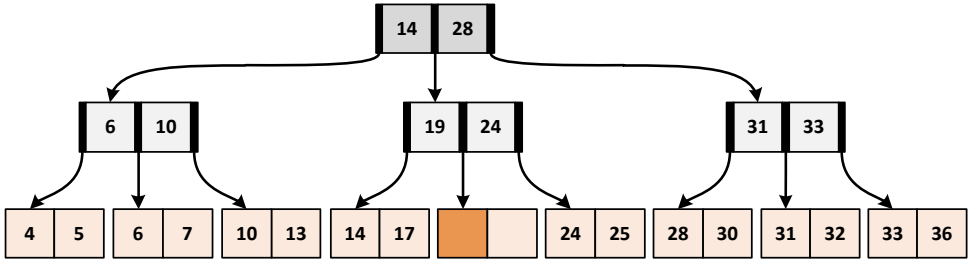
PO operaciji brisanja 21

ISAM indeks naloga 1.b (7)

- **Izbriši** zapise s ključi **22**, **20**, **21**, **19** in **31**.



PRED operacijo



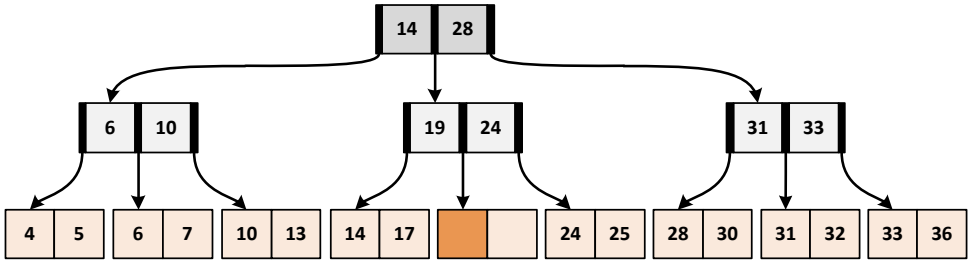
PO operaciji brisanja 19

Izbrisali smo zadnji element osnovne strani, vendar jo kljub temu ne izbrišemo, ampak ohranimo za prihodnje operacije.

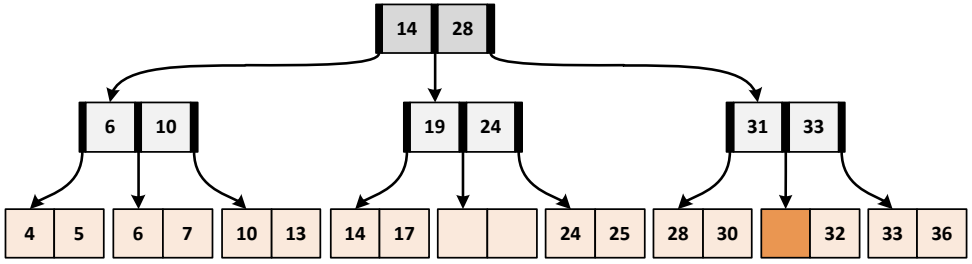
1

ISAM indeks naloga 1.b (8)

- **Izbriši** zapise s ključi **22**, **20**, **21**, **19** in **31**.



PRED operacijo

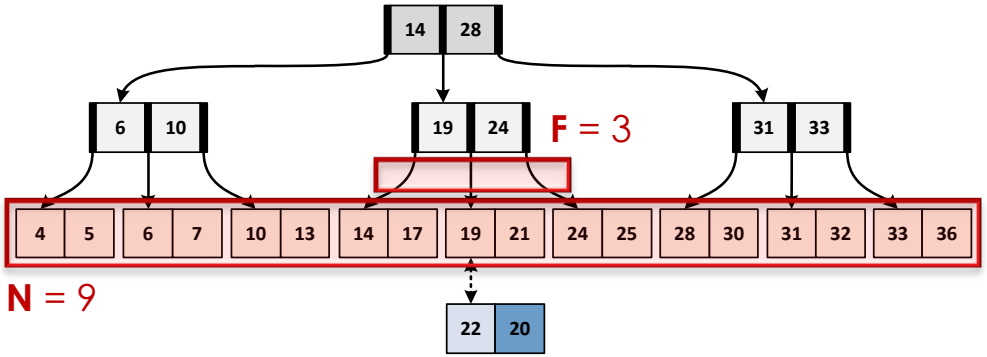


PO operaciji brisanja 31

Ker je ISAM indeks statičen v indeksu ostane ključ 31, kljub temu, da ga med podatki ni več.

ISAM indeks naloga 1.c (9)

- Kolikšno je **število I/O operacij** za **branje** določenega zapisa?
  - **št. I/O operacij** = število nivojev drevesa = $\log_F N$
    - **F** … število otrok vsake indeksne strani
    - **N** … število primarnih listov



št. IO operacij = $\log_F N = \log_3 9 = \mathbf{2}$

Upoštevajmo še prelivne strani

št. IO operacij = $2 + 1 = \underline{\mathbf{3}}$

ISAM indeks naloga 2 (10)

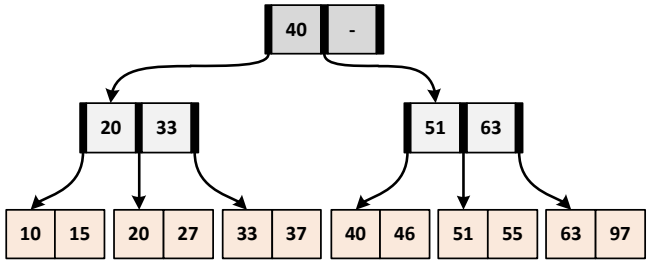
- Recimo da podatkovna baza podpira ISAM in B+ indeks. **Ali je** sploh kdaj **smiselno uporabiti statični indeks ISAM**, če imamo na voljo tudi dinamični B+ indeks?

ISAM indeks naloga 2 (11)

- **Rešitev**
  - ISAM brez prelivnih strani je **hitrejši** od B+ indeksa **pri operacijah dodajanja in brisanja**, ker se pri tem pristopu indeksna vozlišča samo bere in se dinamično ne prilagaja.
    - Statični indeks zgradimo ob **upoštevanju dodatnega praznega prostora** (20%) za vstavljanja.
    - Če se sistem periodično ugaša (npr. nedelovni čas od 15:00 do 7:00), lahko takrat ponovno zgradimo statični indeks. Če so posodabljanja redka in vnaprej znana, potem je smiselno uporabljati ISAM indeks.

ISAM in B+ indeks naloga 3 (12)

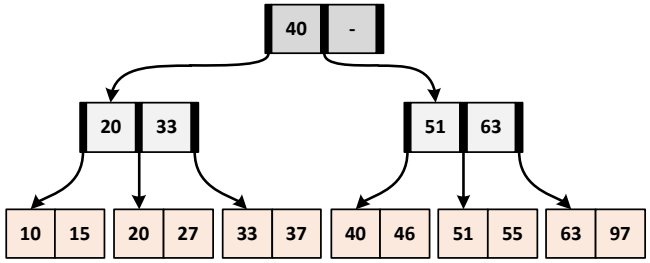
- Nad osnovno datoteko zgradimo dva indeksa (ISAM in B+), ki sta na začetku enaka.



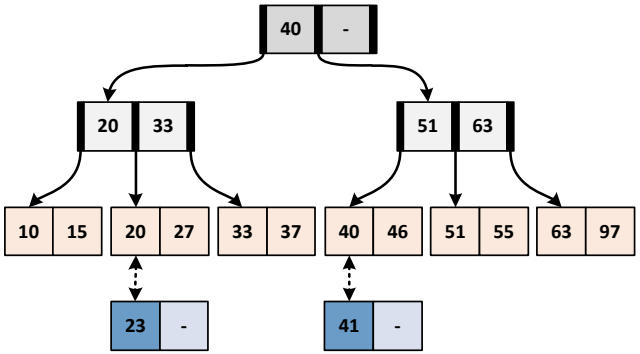
- Kako se spremenita indeksa, če najprej **dodamo** zapis s ključem **41** in nato še zapis s ključem **23**?

ISAM in B+ indeks naloga 3 (13)

- **ISAM indeks**: **dodamo** zapisa s ključema **41** in **23**



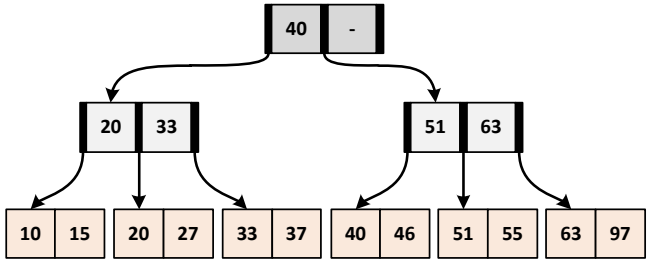
PRED operacijo



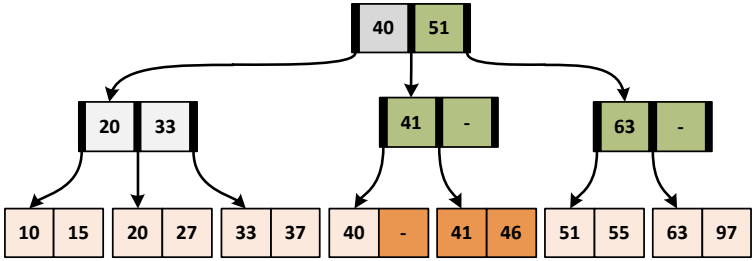
PO operacijah dodajanja

ISAM in B+ indeks naloga 3 (14)

- **B+ indeks**: **dodamo** zapisa s ključema **41** in **23**



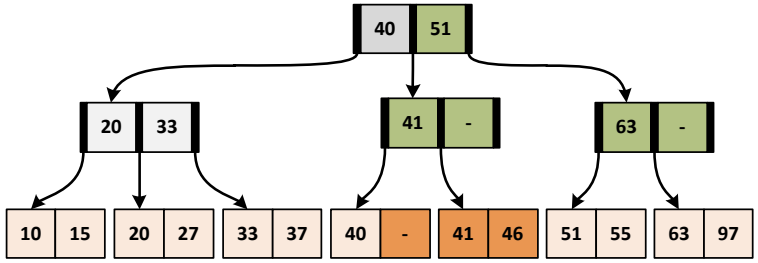
PRED operacijo



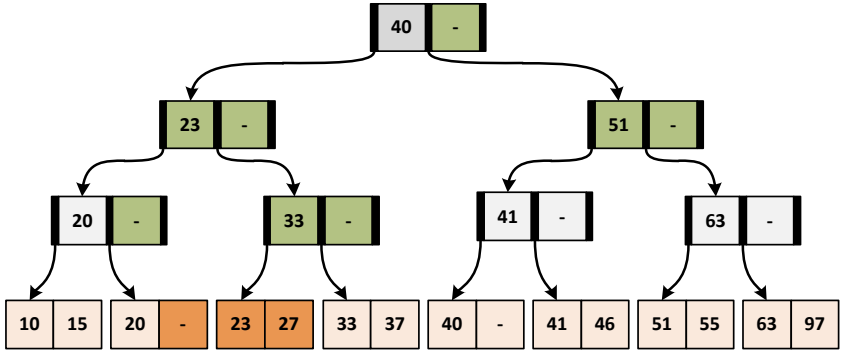
PO operaciji dodajanja 41

ISAM in B+ indeks naloga 3 (15)

- **B+ indeks**: **dodamo** zapisa s ključema **41** in **23**



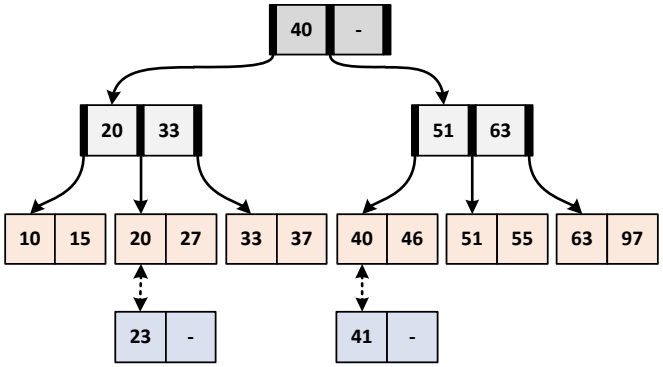
PRED operacijo



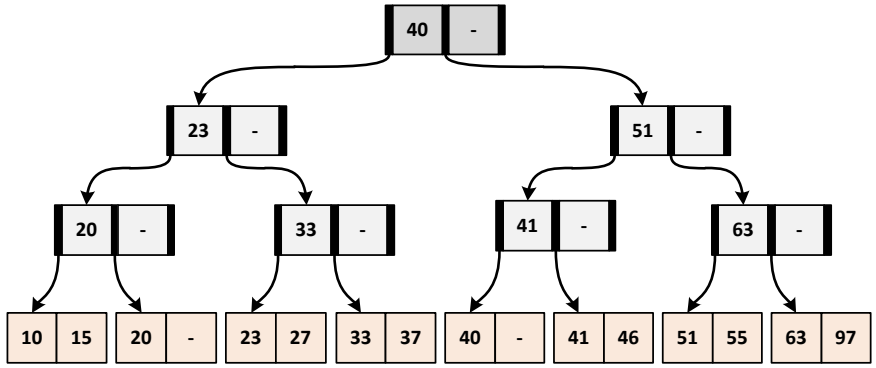
PO operaciji dodajanja 23

ISAM in B+ indeks naloga 3 (16)

- ISAM in B+ sta dinamična indeksa?
- ISAM indeks predstavlja uravnoteženo drevo?
- ISAM indeks je pri branju podatka s ključem 41 hitrejši od B+ indeks?



ISAM indeks

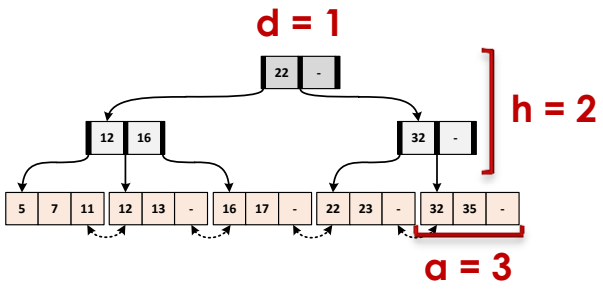


B+ indeks

Indeksiranje

Izračun števila kazalcev in zapisov.

Izračun števila kazalcev in zapisov



- **Število kazalcev** na h nivoju je odvisno od razvejanosti F:
  - $F = 2d + 1$ … razvejanost drevesa
  - h … višina drevesa (brez listov)
  - a … število zapisov v listih
  - **Max.** število zapisov: $\boldsymbol{a} \cdot \boldsymbol{(2d+1)^h}$
  - **Min.** število zapisov: $\frac{\boldsymbol{a}}{\boldsymbol{2}} \cdot \boldsymbol{2(d+1)^{h-1}}$
    - Število kazalcev za polovično zasedeno vozlišče

Število kazalcev

Naloga 4 (17)

- Koliko zapisov v osnovni datoteki lahko indeksiramo z B+ indeksom reda 50, ki ima 3 nivoje indeksnih blokov, bloki v gostem indeksu pa so veliki 10 indeksnih zapisov?

$d = 50$
$h = 3$
$a = 10$

Max. število kazalcev
$= (2d + 1)^h = (2 \cdot 50 + 1)^3 = 101^3 = 1.030.301$

Max. število zapisov = a · max. št. kazalcev
$= 10 \cdot 1.030.301 = \mathbf{10.303.010}$

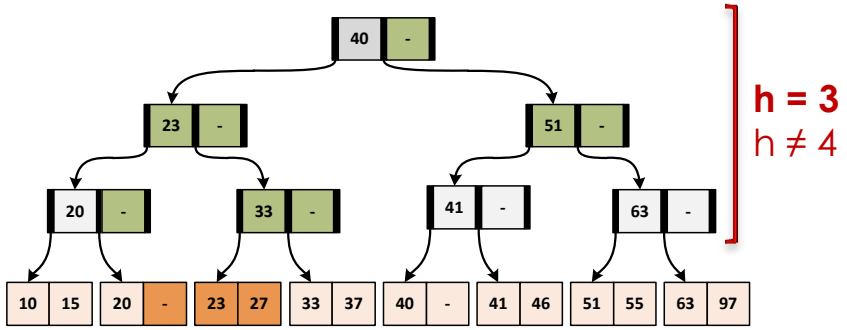
Naloga 5 (18)

- Koliko največ listov lahko imamo v B+ indeksu, če je red drevesa 20 in imamo 3 nivoje v B+ delu drevesa?

$d = 20$
$h = 3$

Max. število listov
$= (2d + 1)^h = (2 \cdot 20 + 1)^3 = 41^3 = \mathbf{68.921}$

Naloga 6 (19)



- Osnovna datoteka je indeksirana z B+ indeksom z redom 100, višina pa je 4 nivoje, skupaj z listi. List vsebuje 20 indeksnih zapisov. Izračunajte min. in max. število zapisov v osnovni datoteki!

$d = 100$
$\boldsymbol{h = 3}$
$a = 20$

Max. število zapisov
$= a \cdot (2d + 1)^h = 20 \cdot (2 \cdot 100 + 1)^3 = 20 \cdot 201^3 = \mathbf{162.412.020}$

Min. število zapisov
$= \frac{a}{2} \cdot 2(d + 1)^{h-1} = 20 \cdot (100 + 1)^2 = 20 \cdot 101^2 = \mathbf{204.020}$

Indeksiranje

Bitni indeks.

Bitni indeks

- Bitni indeks se ponavadi uporablja v **podatkovnih skladiščih**.
- Bitni indeks je zelo učinkovit, če ima atribut **majhno število možnih vrednosti**.
- Od B+ indeksa je **boljši pri iskanju po več predikatih**, pod pogojem da za njih obstaja bitni indeks.

Bitni indeks primer (1)

- Imamo tabelo **Staff**.

staffNo	fName	position	salary	branchNo
SG14	David	Manager	18.000	B003
SG37	Ann	Assistant	12.000	B003
SL21	John	Supervisor	30.000	B005
SL41	Julie	Assistant	9.000	B005
SF56	Tim	Manager	32.000	B005

- Kreirajmo bitni indeks po polju **position**!

Manager	Assistant	Supervisor
1	0	0
0	**1**	0
0	0	**1**
0	**1**	0
1	0	0

1. Pogledamo različne vrednosti polja position in te **vrednosti postanejo novi stolpci**.
2. Število vrstic v indeksu je enako številu vrsti tabele.
3. Za vsako vrednost preverimo ali se nahaja (1) v izbrani vrstici.

Bitni indeks primer (2)

- Imamo tabelo **Staff**.

staffNo	fName	position	salary	branchNo
SG14	David	Manager	18.000	B003
SG37	Ann	Assistant	12.000	B003
SL21	John	Supervisor	30.000	B005
SL41	Julie	Assistant	9.000	B005
SF56	Tim	Manager	32.000	B005

- Kreirajmo še bitni indeks po polju **branchNo**!

B003	B005
1	0
1	0
0	**1**
0	**1**
0	**1**