

ALGORITMI IN PODATKOVNE STRUKTURE 1



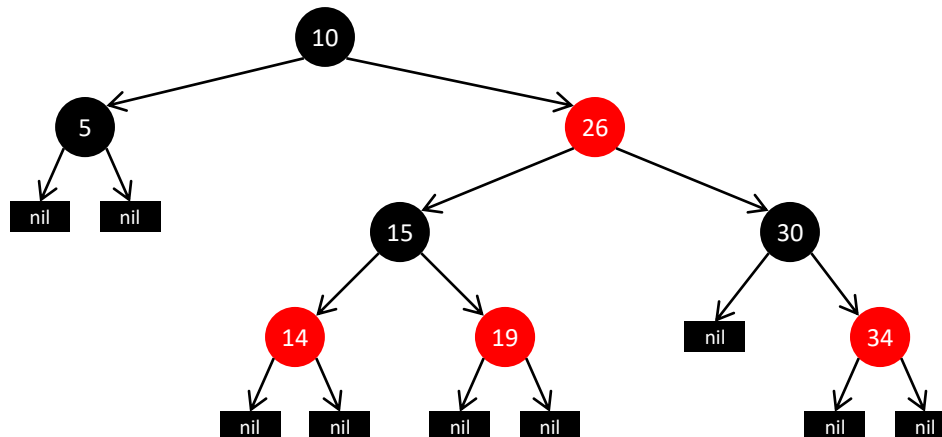
9. laboratorijske vaje

Rdeče-črno drevo, AVL drevo, B-drevo, kopica

RDEČE-ČRNO DREVO

Binarno iskalno drevo za katerega velja:

- vsako vozlišče je bodisi **rdeče** bodisi **črne** barve
- **rdeče** vozlišče ima lahko samo **črna** sinova
- za **vsako** vozlišče velja, da **vsaka pot** od vozlišča do praznega poddrevesa (nil) vsebuje **enako** število **črnih** vozlišč



DODAJANJE ELEMENTA V RDEČE-ČRNO DREVO



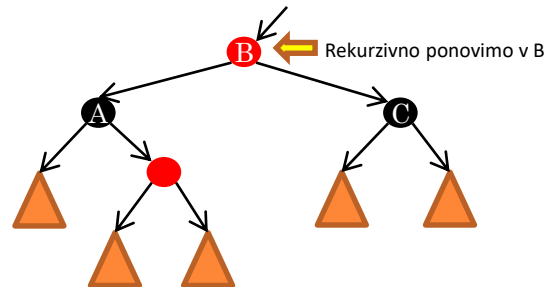
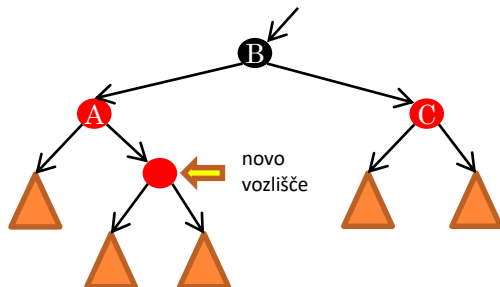
1. Element dodamo v list drevesa kot pri navadnem BST.
2. Dodano vozlišče (list) pobarvamo **rdeče**.
3. Če je oče dodanega lista **rdeče** vozlišče, je potrebno drevo popraviti.

PRAVILA ZA DODAJANJE V RDEČE-ČRNO DREVO

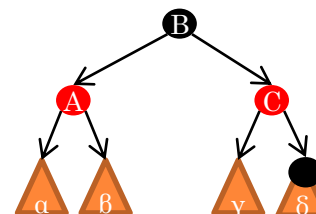
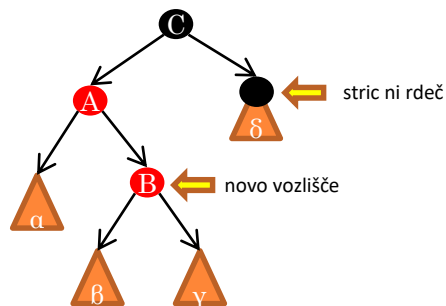
D1



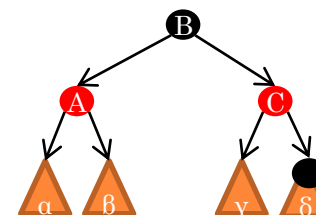
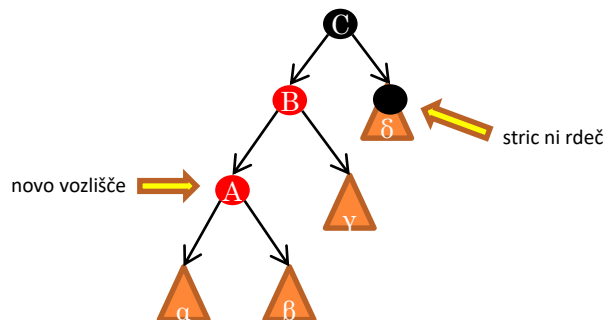
D2



D3



D4



VSTAVLJANJE V RDEČE-ČRNO DREVO

Na začetku je rdeče-črno drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 16.

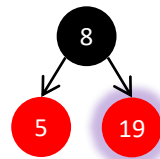
8:



5:



19:



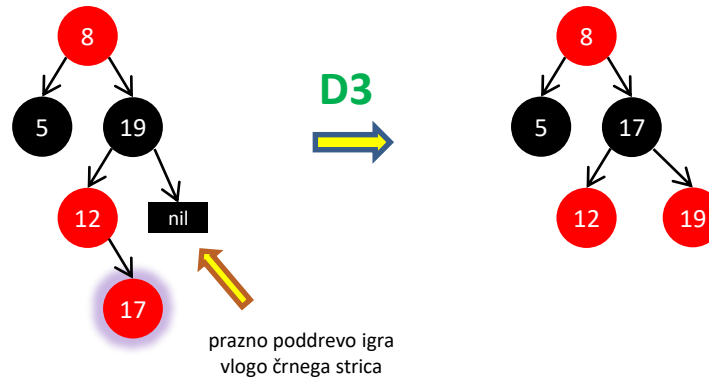
12:



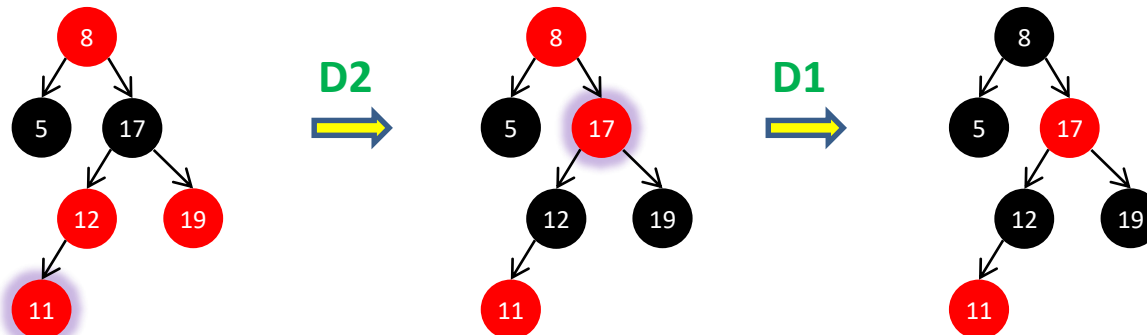
VSTAVLJANJE V RDEČE-ČRNO DREVO

Na začetku je rdeče-črno drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 16.

17:



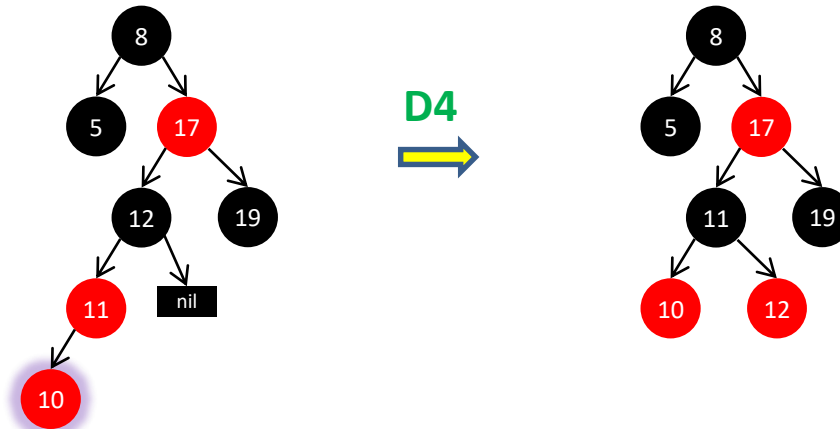
11:



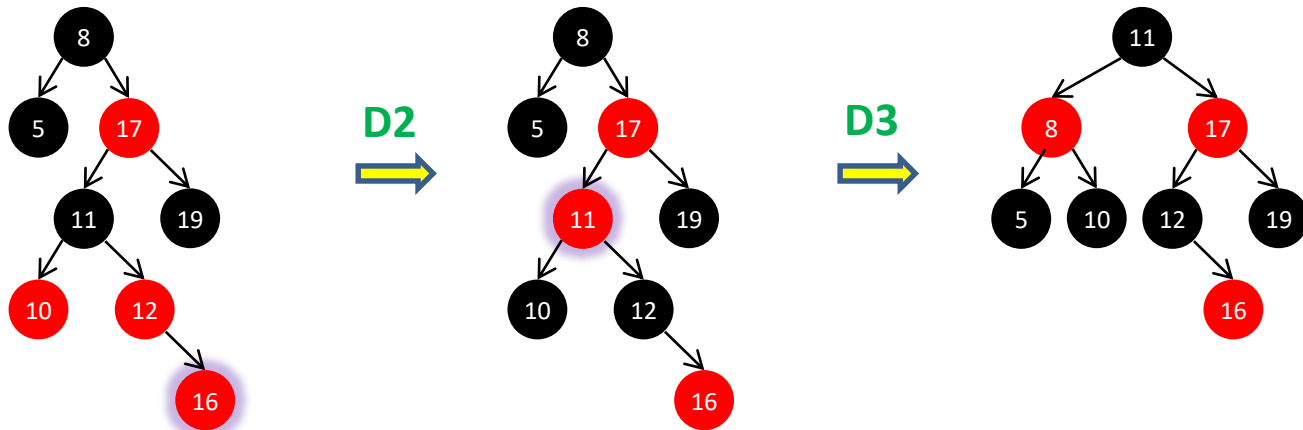
VSTAVLJANJE V RDEČE-ČRNO DREVO

Na začetku je rdeče-črno drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 16.

10:



16:



BRISANJE ELEMENTA IZ RDEČE-ČRNEGA DREVESA

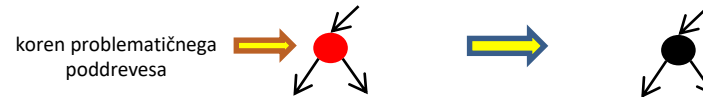


1. Element zbrisemo iz drevesa kot pri navadnem BST.
2. Če je zbrisano **rdeče** vozlišče končamo;
Če je zbrisano **črno** vozlišče, je potrebno drevo popraviti.

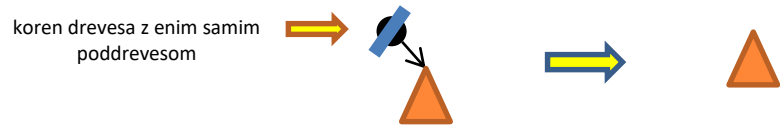
PRAVILA ZA BRISANJE IZ RDEČE-ČRNEGA DREVEVA

Če je zbrisano črno vozlišče drevo popravljamo...

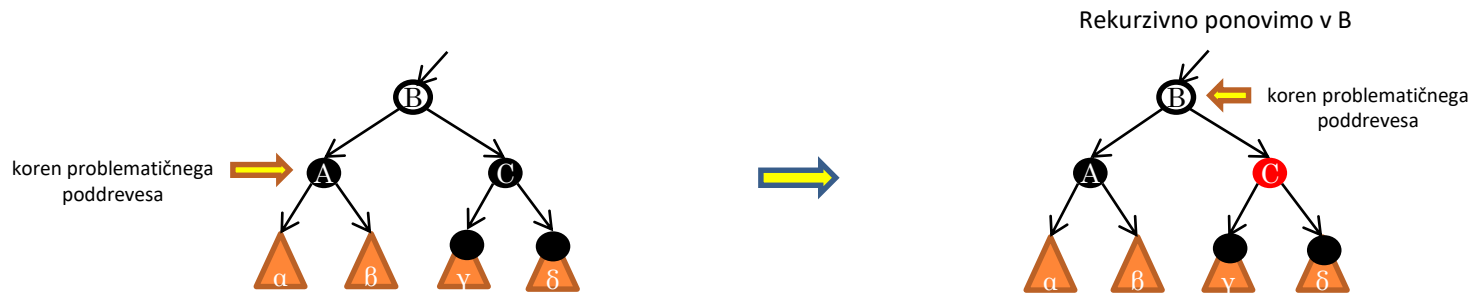
B1



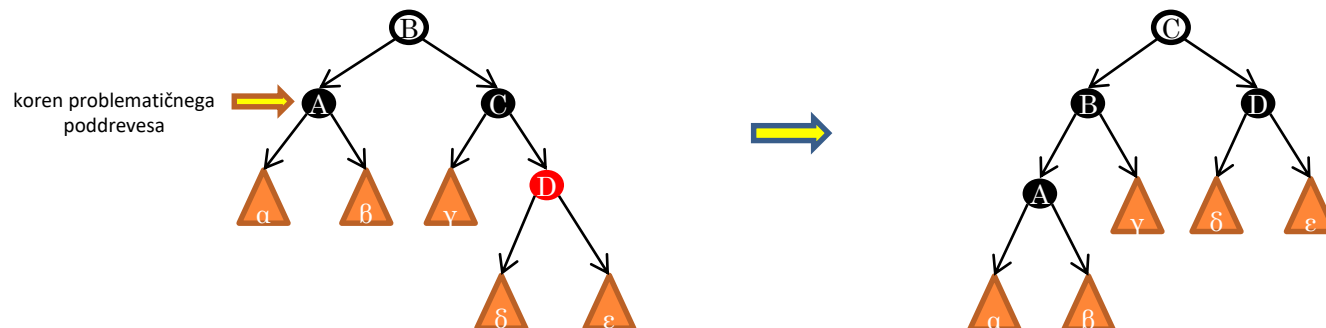
B2



B3



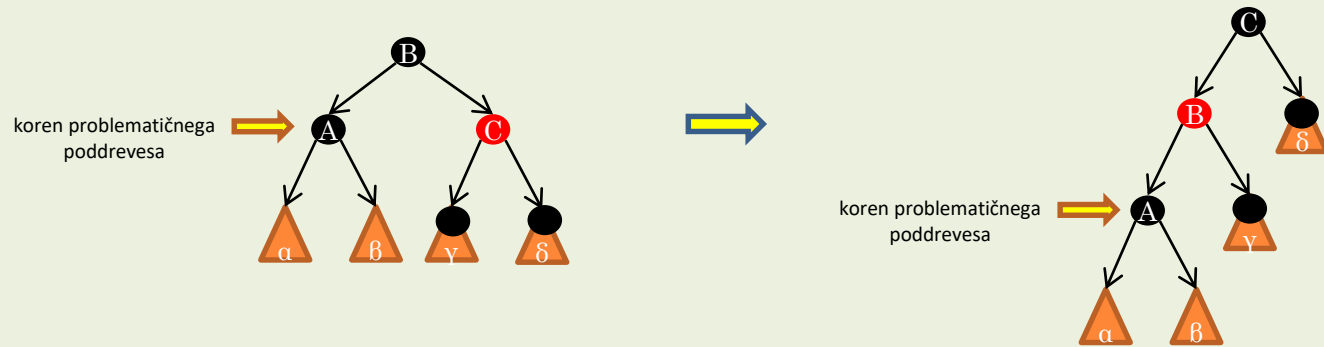
B4



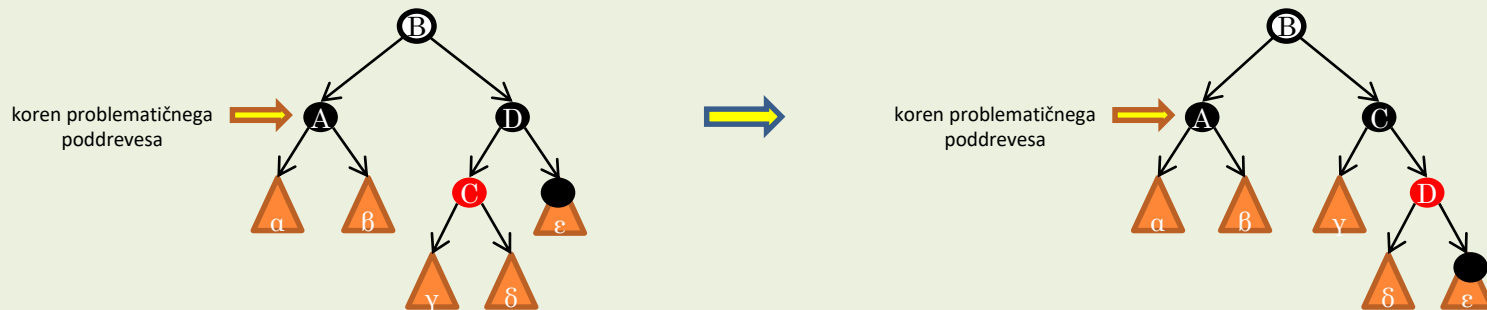
PRAVILA ZA BRISANJE IZ RDEČE-ČRNEGA DREVEVA

Predpriprava...

P1



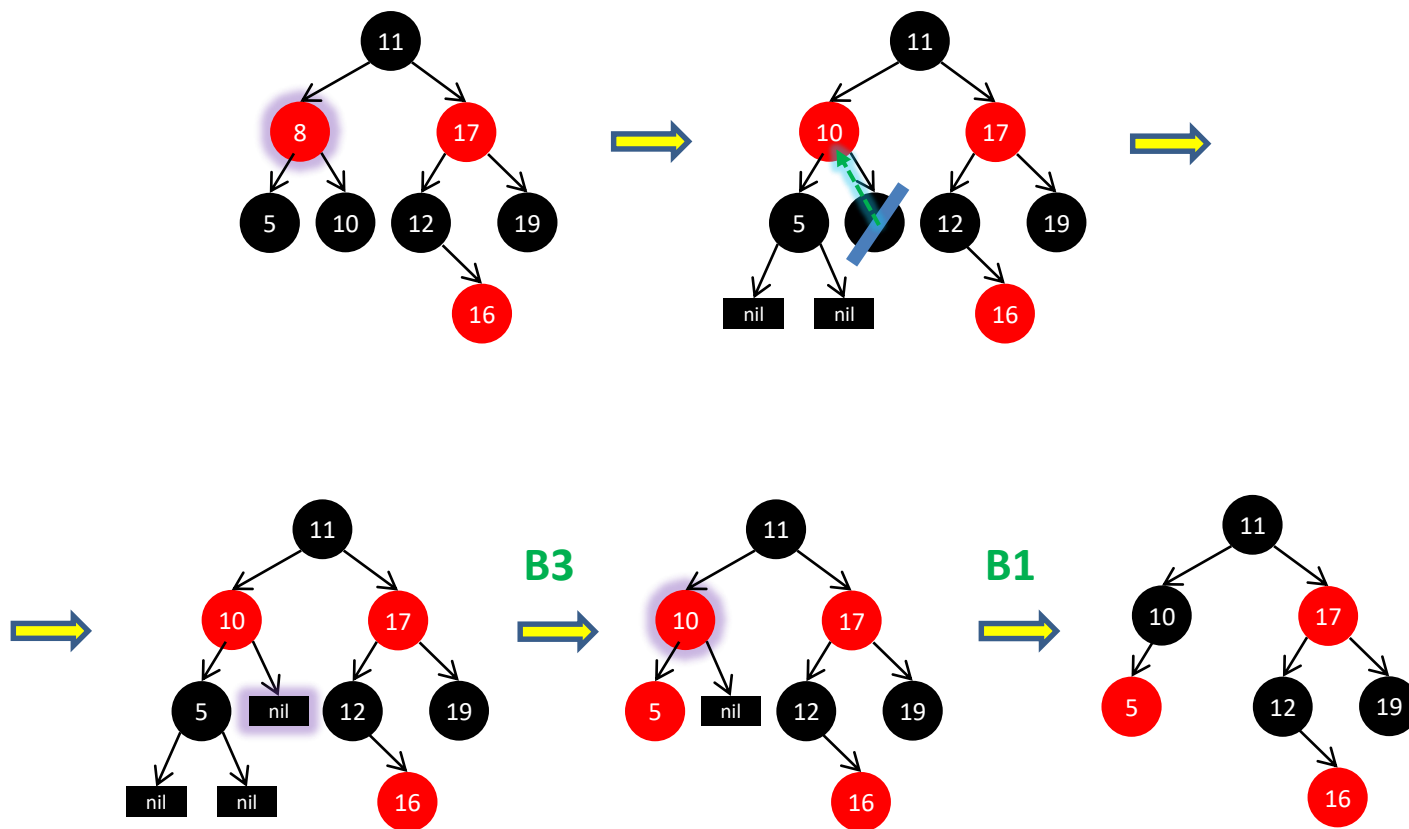
P2



BRISANJE IZ RDEČE-ČRNEGA DREVEVA

Odstranite iz drevesa elemente: 8, 10 in 5.

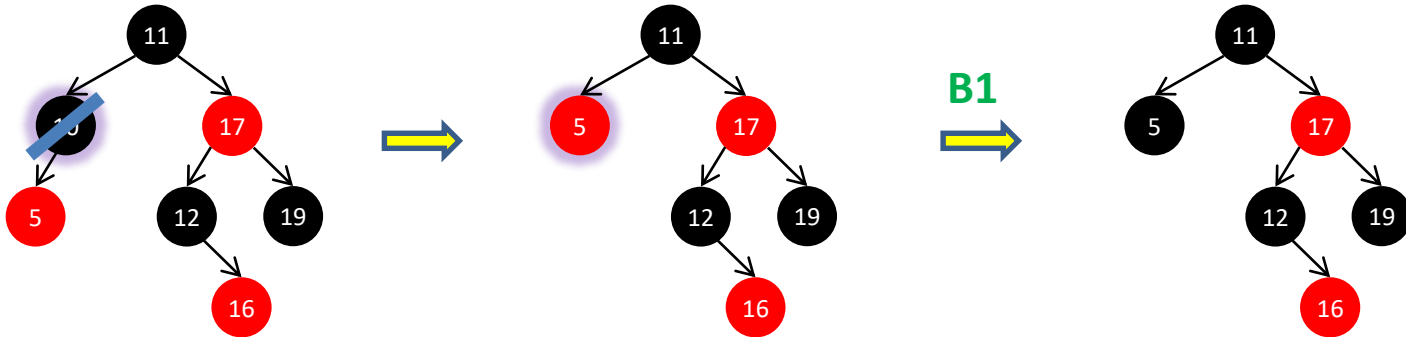
8:



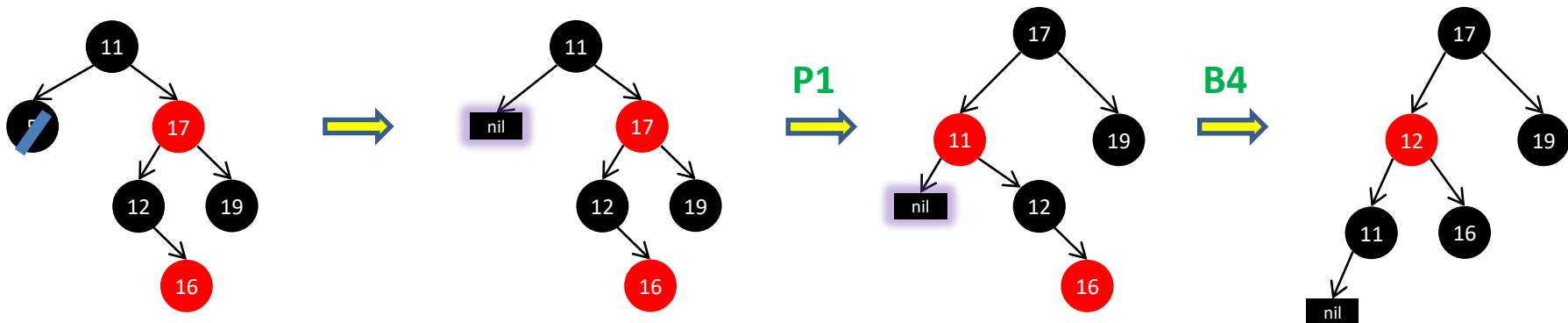
BRISANJE IZ RDEČE-ČRNEGA DREVEVA

Odstranite iz drevesa elemente: 8, 10 in 5.

10:

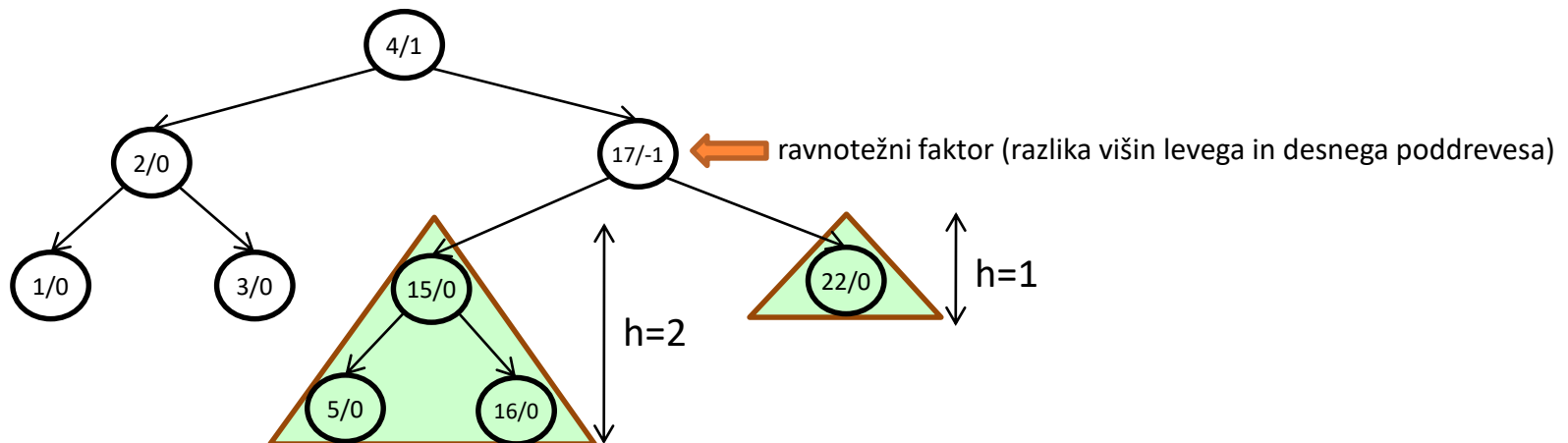


5:



AVL-DREVO

- poimenovano po avtorjih Adelson-Velskii in Landis
- delno poravnano binarno iskalno drevo
- za vsako vozlišče velja, da se višini obeh poddreves razlikujeta največ za 1

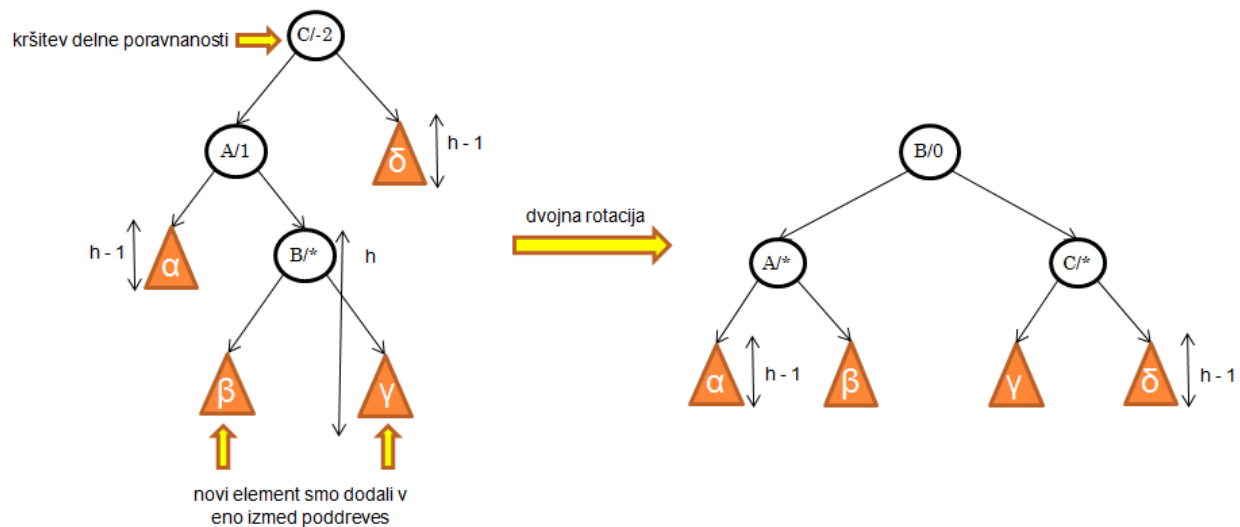
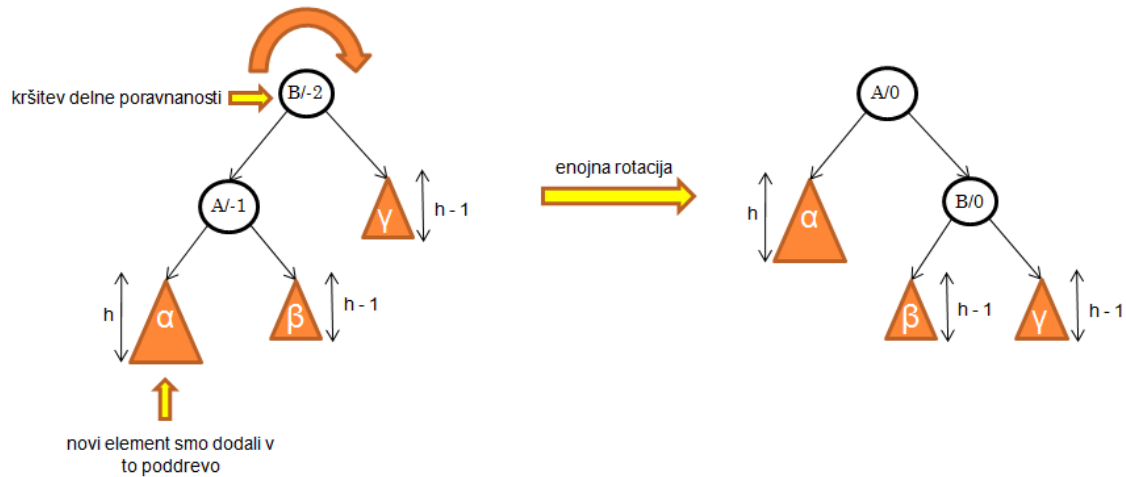


DODAJANJE ELEMENTA V AVL-DREVO



1. Element dodamo v list drevesa kot pri navadnem BST.
2. Preverimo ravnotežni faktor vseh vozlišč na poti navzgor od vstavljenega lista do korena drevesa.
 - če je absolutna vrednost ravnotežnega faktorja večja kot 1, je potrebno drevo popravljati
 - v najslabšem primeru je potrebno popravljati ravnotežni faktor vse do korena - ko pa pride do rotacije (enojne ali dvojne) je stvar zaključena

DODAJANJE ELEMENTA V AVL-DREVO



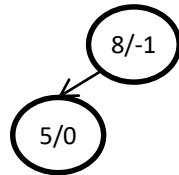
VSTAVLJANJE V AVL-DREVO

Na začetku je AVL-drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 9.

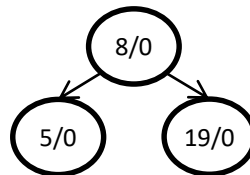
8:



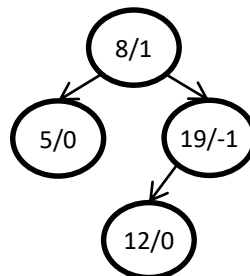
5:



19:



12:



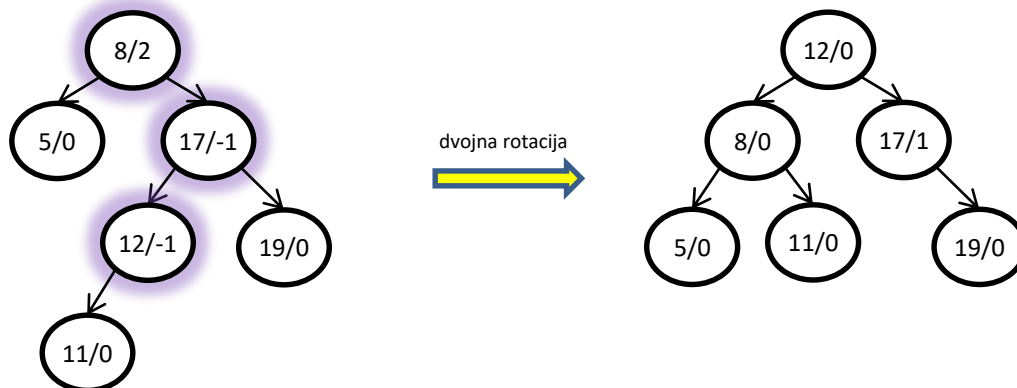
VSTAVLJANJE V AVL-DREVO

Na začetku je AVL-drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 9.

17:



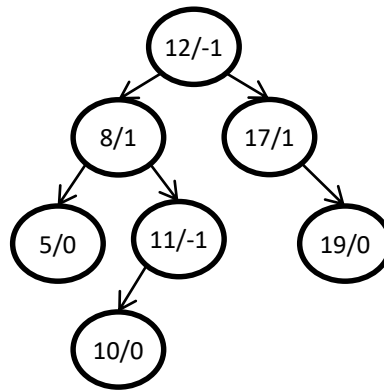
11:



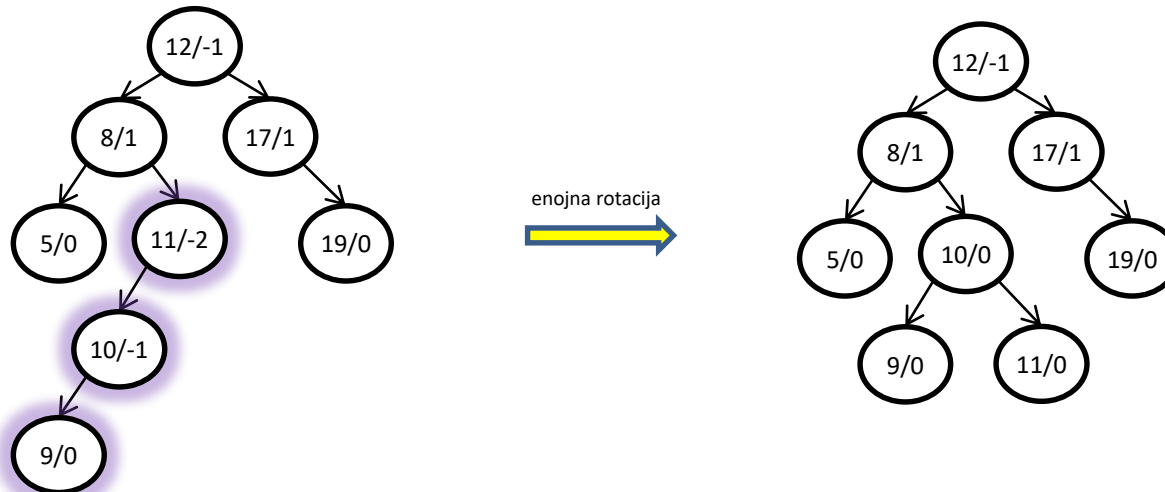
VSTAVLJANJE V AVL-DREVO

Na začetku je AVL-drevo prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11, 10 in 9.

10:



9:

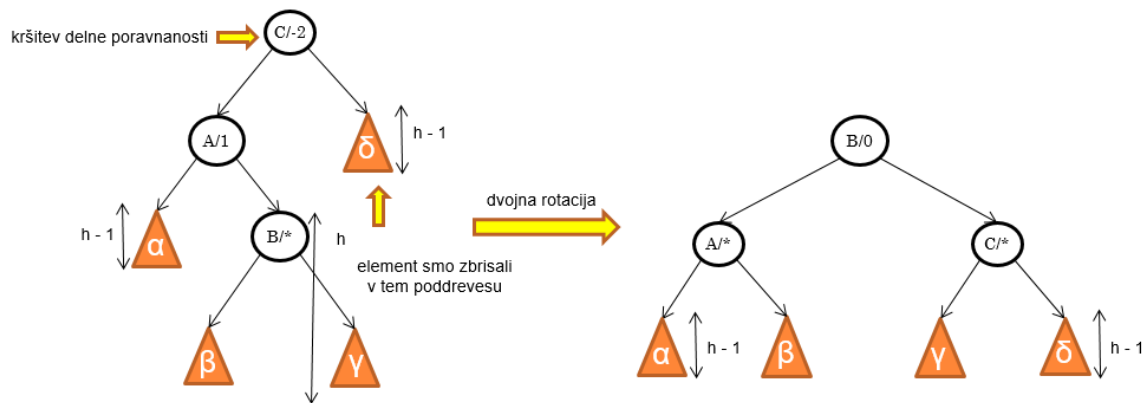
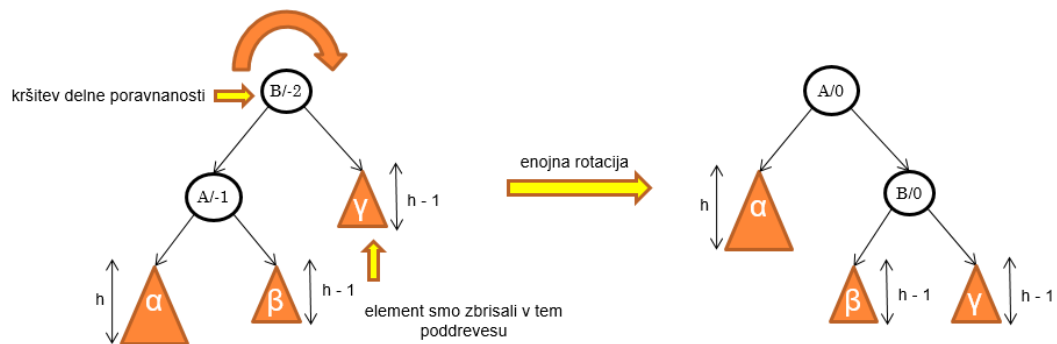
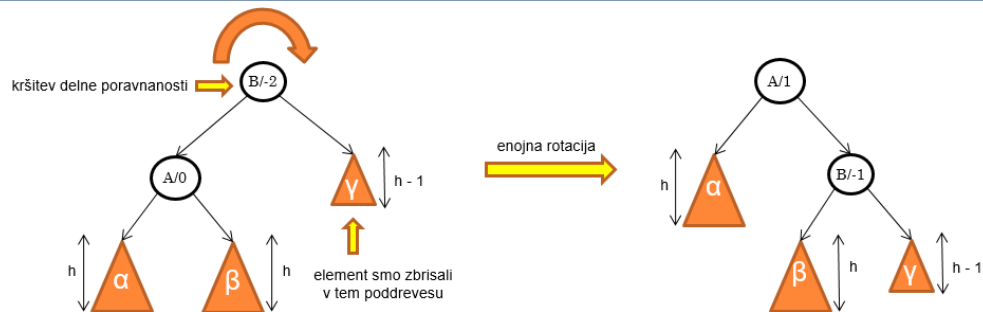


BRISANJE ELEMENTA IZ AVL-DREVEESA



1. Element brišemo iz drevesa kot pri navadnem BST:
2. Preverimo ravnotežni faktor vseh vozlišč na poti navzgor od očeta dejansko zbrisanega vozlišča do korena drevesa.
 - če je absolutna vrednost ravnotežnega faktorja večja kot 1, je potrebno drevo popravljati
 - v najslabšem primeru je potrebno po celi poti od zbrisanega vozlišča do korena poravnati drevo

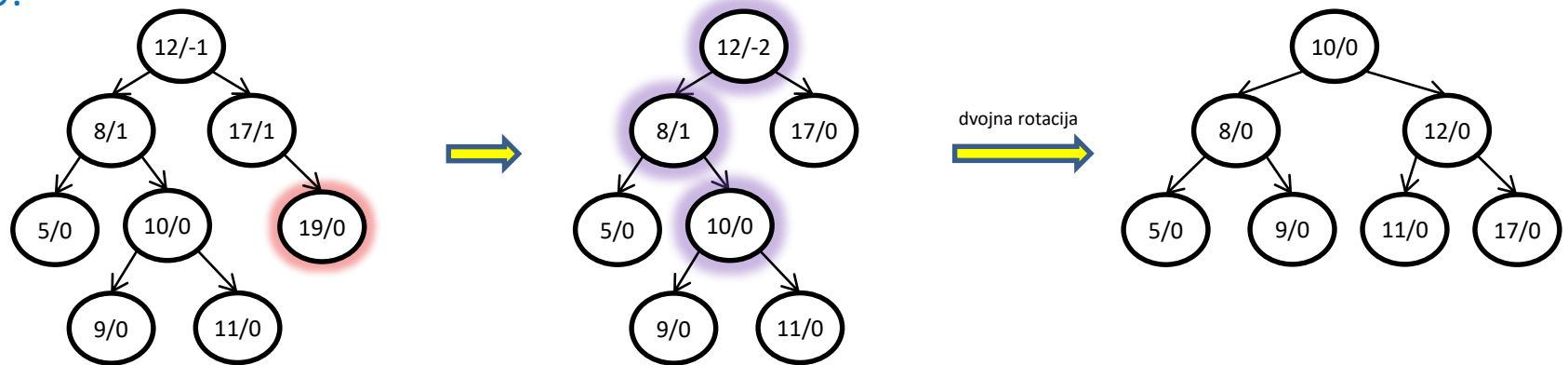
BRISANJE ELEMENTA IZ AVL-DREVEESA



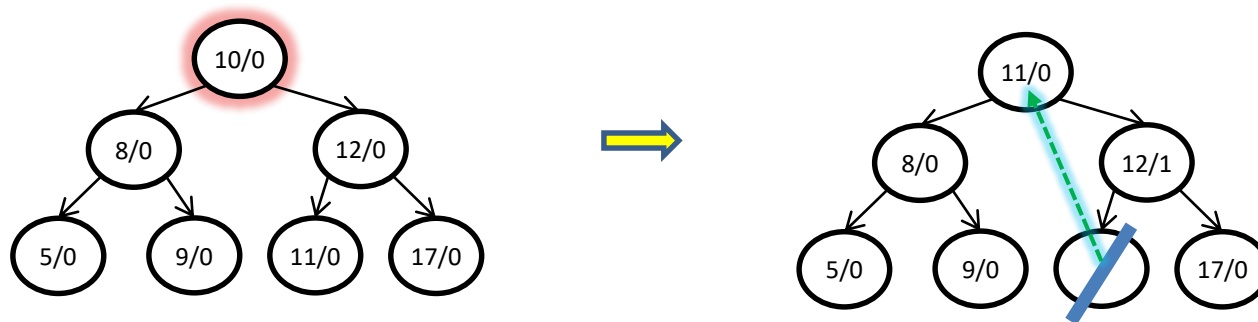
BRISANJE IZ AVL-DREVEESA

Odstranite iz drevesa elemente: 19, 10, 12, 9 in 17.

19:



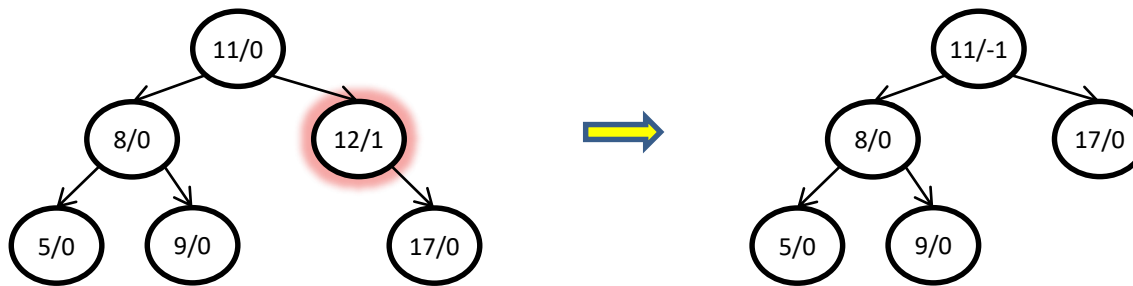
10:



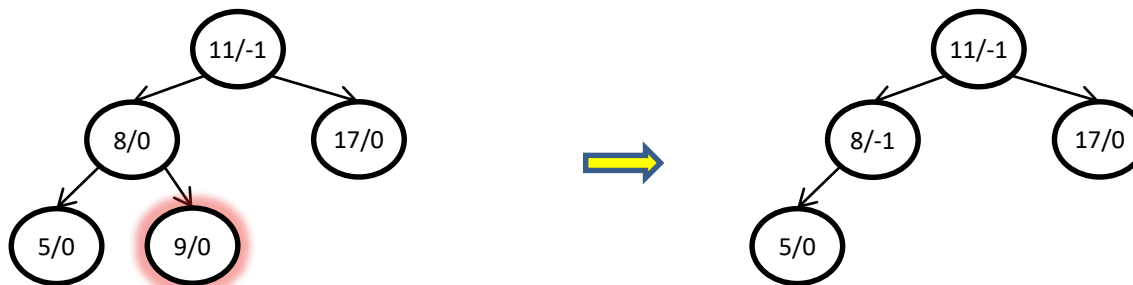
BRISANJE IZ AVL-DREVEESA

Odstranite iz drevesa elemente: 19, 10, 12, 9 in 17.

12:



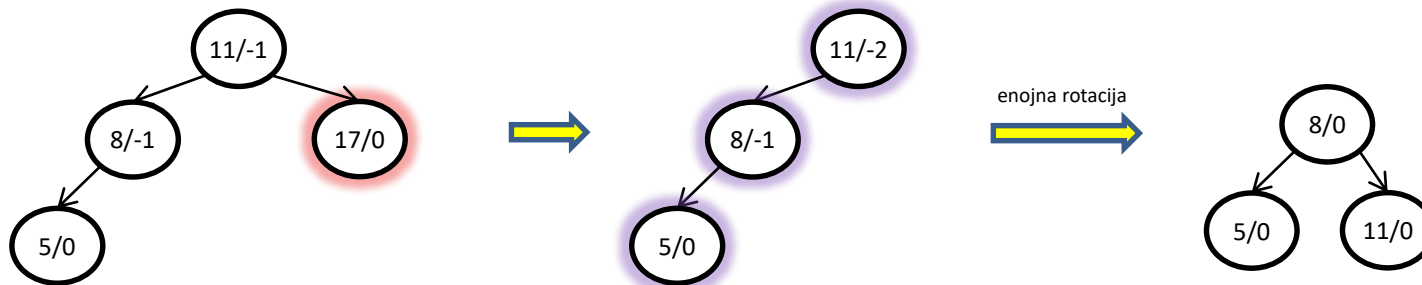
9:



BRISANJE IZ AVL-DREVEESA

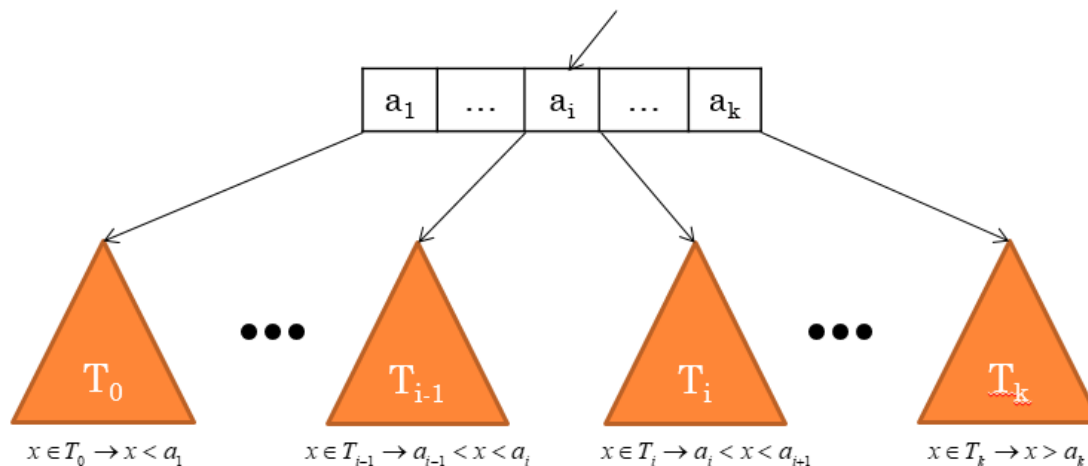
Odstranite iz drevesa elemente: 19, 10, 12, 9 in 17.

17:



B-DREVO

- B-drevo je popolnoma poravnano iskalno drevo – vsi listi so na istem nivoju
- vsako notranje vozlišče B-drevesa reda m ima lahko od $\lceil m/2 \rceil$ do m sinov ter en ključ manj kot ima sinov
- izjema je koren drevesa – v primeru, da je koren notranje vozlišče, ima lahko od 2 do m sinov



ISKANJE ELEMENTA V B-DREVESU



Iskanje elementa je posplošitev iskanja v navadnih binarnih iskalnih drevesih:

1. iskanje začnemo v vozlišču, ki je koren drevesa
2. iskani element zaporedno primerjamo z elementi v vozlišču, dokler:
 - ne naletimo na iskani element
 - ne naletimo na večji element in se iskanje rekurzivno nadaljuje v poddrevesu z istim indeksom
 - ne pregledamo zadnjega elementa in se iskanje rekurzivno nadaljuje v zadnjem poddrevesu

DODAJANJE ELEMENTA V B-DREVO

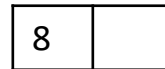
Element dodajamo v ustrezno notranje vozlišče največje globine:

1. če opazovano vozlišče vsebuje manj kot $m-1$ elementov, dodamo element na ustrezno mesto in končamo
2. če v opazovanem vozlišču ni prostora, ga razbijemo na dva vozlišča (novo levo in novo desno vozlišče):
 - določimo sredinski element izmed elementi v vozlišču in novim elementom, ki ga dodajamo
 - elemente, ki so manjši od sredinskega elementa, dodamo v novo levo vozlišče
 - elemente, ki so večji od sredinskega elementa, dodamo v novo desno vozlišče
 - sredinski element rekurzivno dodamo očetu prejšnjega vozlišča (rekurzija se konča bodisi pri 1. točki, bodisi dobimo nov koren drevesa)

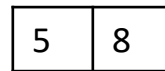
VSTAVLJANJE V B-DREVO

Na začetku je B-drevo reda 3 prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11 in 10.

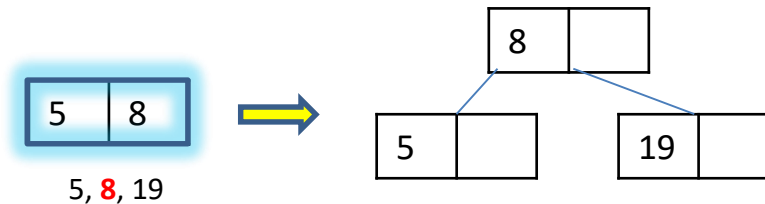
8:



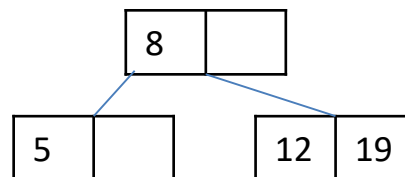
5:



19:



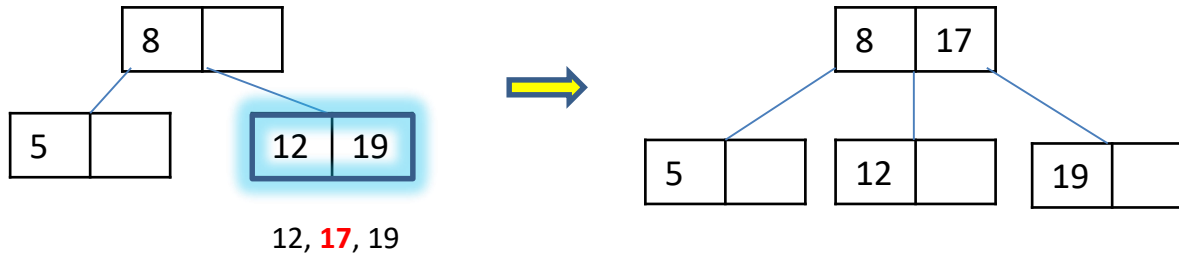
12:



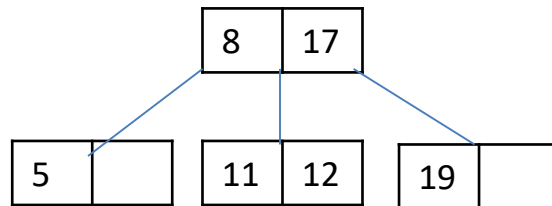
VSTAVLJANJE V B-DREVO

Na začetku je B-drevo reda 3 prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11 in 10.

17:



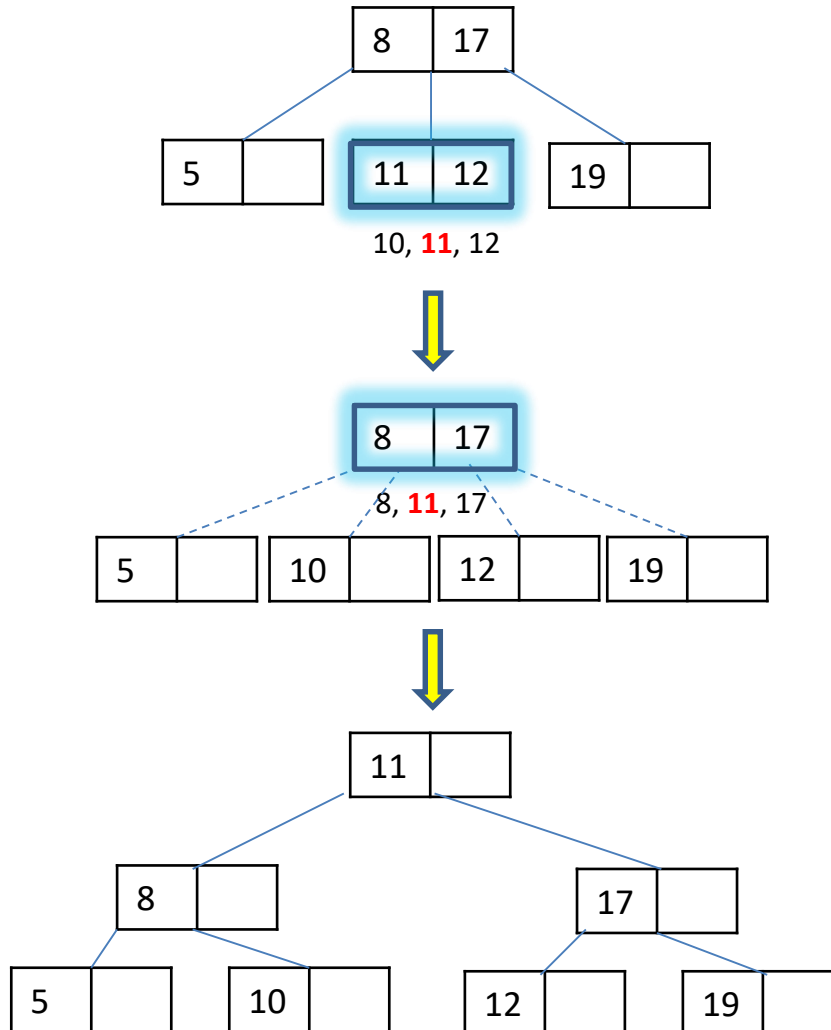
11:



VSTAVLJANJE V B-DREVO

Na začetku je B-drevo reda 3 prazno. Nariši drevo po končanem vstavljanju elementov: 8, 5, 19, 12, 17, 11 in 10.

10:



BRISANJE ELEMENTA IZ B-DREVESA

Vedno brišemo element iz vozlišča na zadnjem nivoju

- pri brisanju elementa, ki ni na zadnjem nivoju, ga nadomestimo s predhodnikom (z največjim elementom ustreznega levega poddrevesa), ki ga dejansko zbrisemo

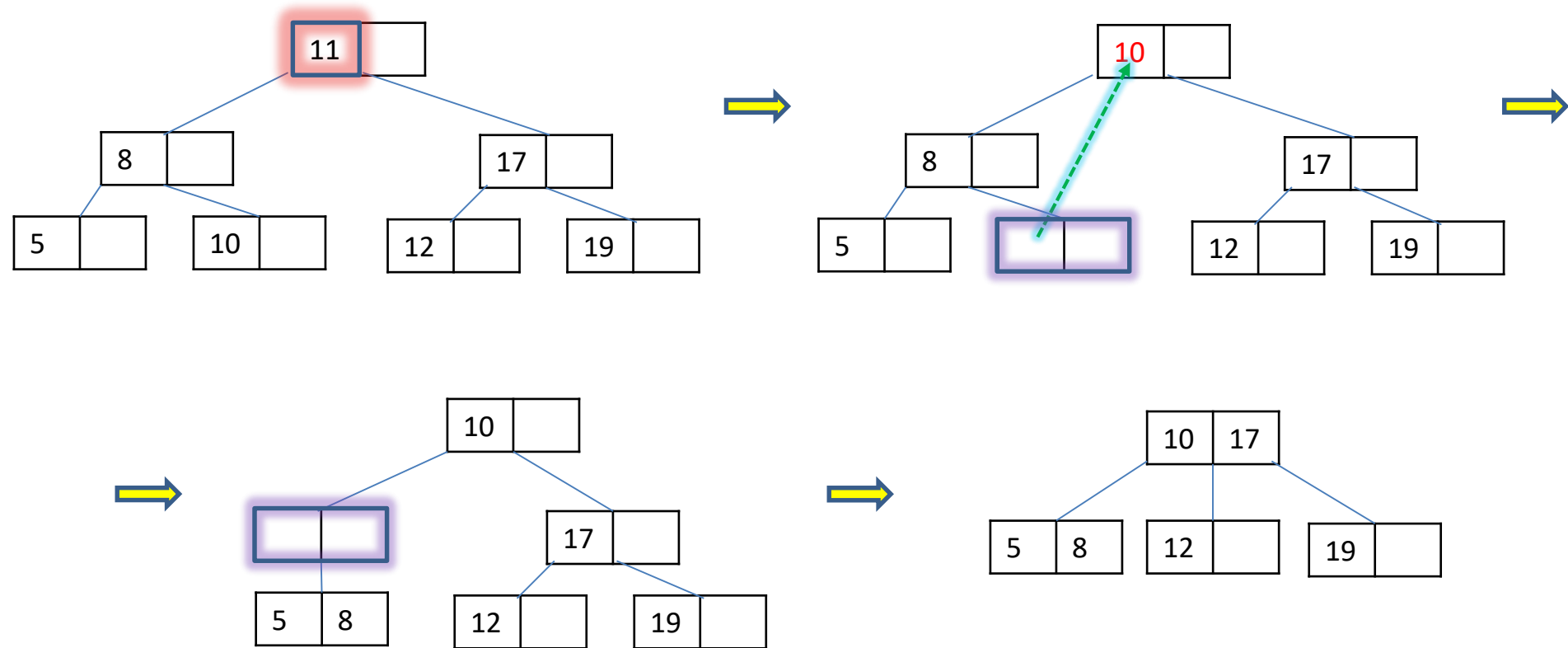
1. če vozlišče, kjer smo element zbrisali, vsebuje dovoljeno število elementov, je postopek končan
2. če vozlišče sedaj vsebuje premalo elementov, potem:
 - a) eden izmed sosednih bratov (levi ali desni) vsebuje dovolj elementov, da si jih razdelita med seboj – v tem primeru se od brata vzame enega ali več elementov skupaj z ustreznimi poddrevesi in z zamenjavo ustreznega elementa pri očetu
 - b) če nobeden od bratov nima dovolj velikega števila elementov, imamo dva brata, ki ju skupaj z ustreznim elementom v očetu lahko združimo v eno vozlišče, postopek pa rekurzivno ponovimo pri očetu

rekurzija se konča bodisi pri 1. ali 2.(a) točki ali pa pri korenu brez elementov (v tem primeru koren zbrisemo)

BRISANJE IZ B-DREVEESA

Odstranite iz drevesa elemente: 11, 12, 19, 8 in 5.

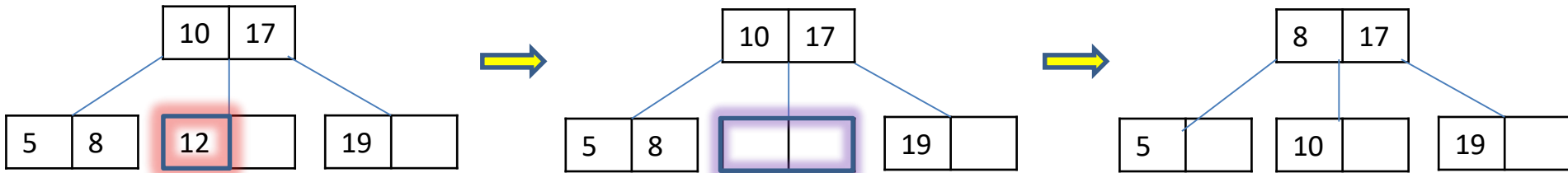
11:



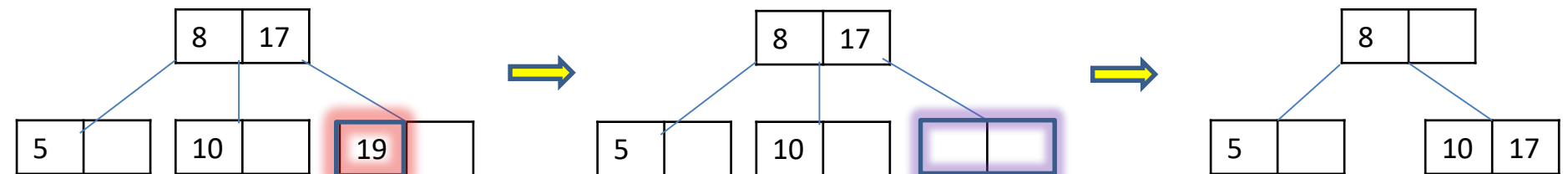
BRISANJE IZ B-DREVEESA

Odstranite iz drevesa elemente: 11, 12, 19, 8 in 5.

12:



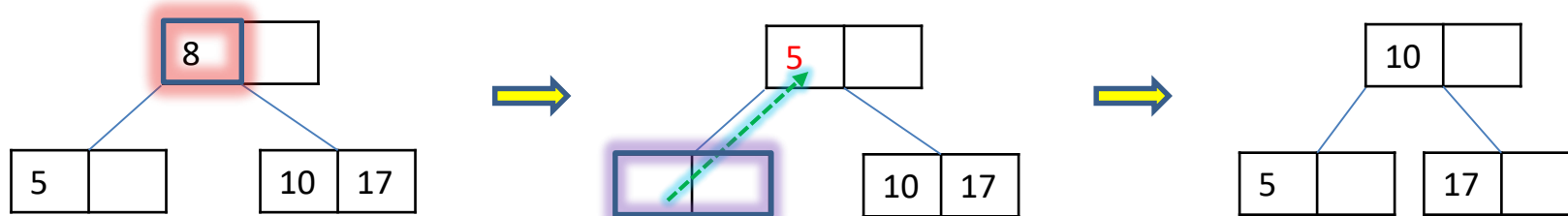
19:



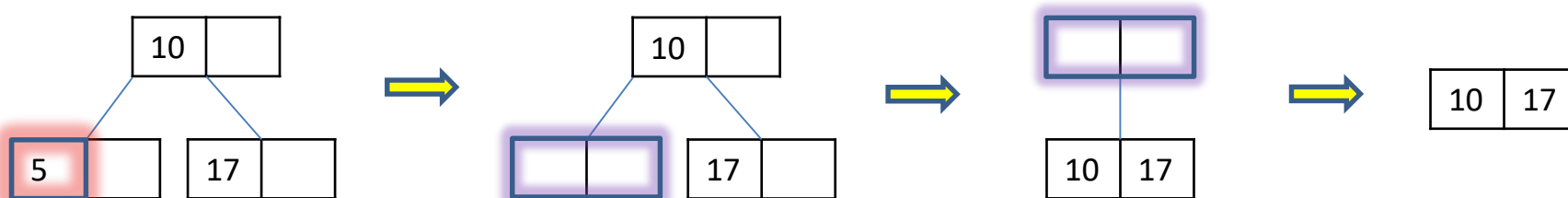
BRISANJE IZ B-DREVEESA

Odstranite iz drevesa elemente: 11, 12, 19, 8 in 5.

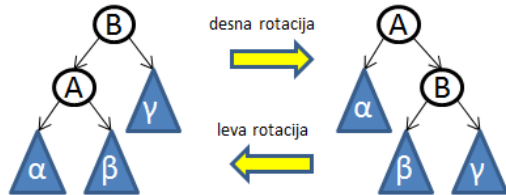
8:



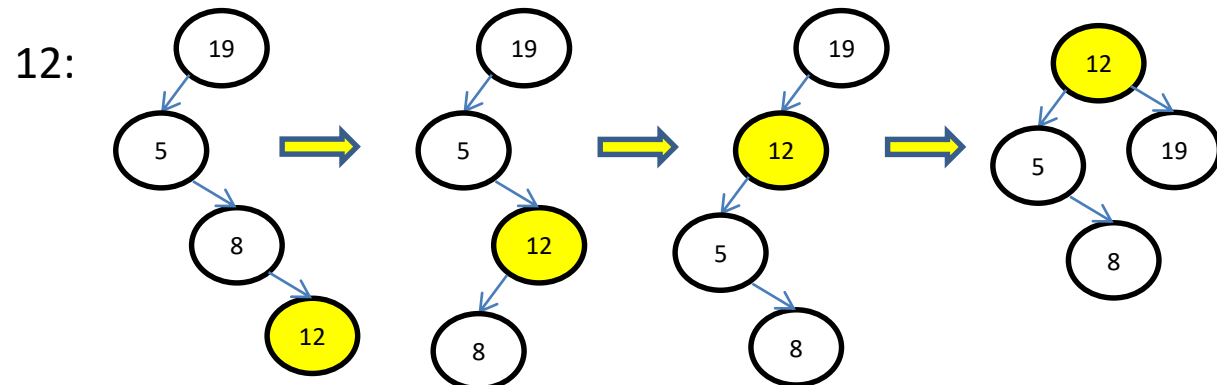
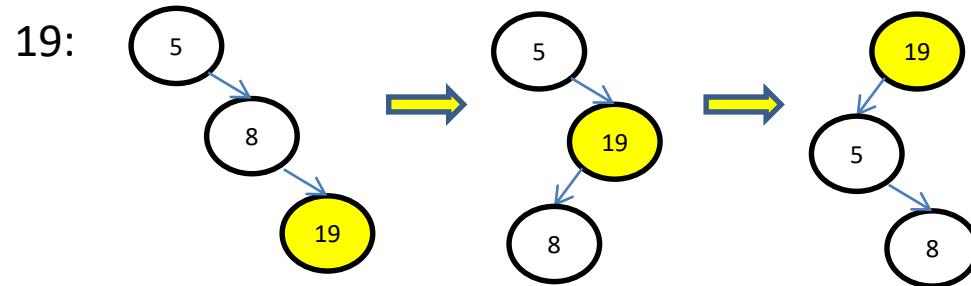
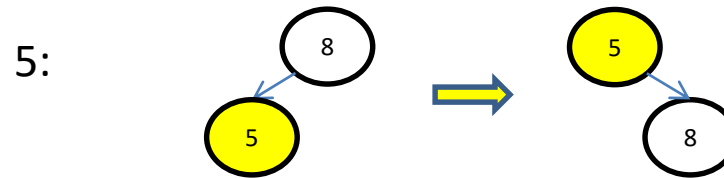
5:



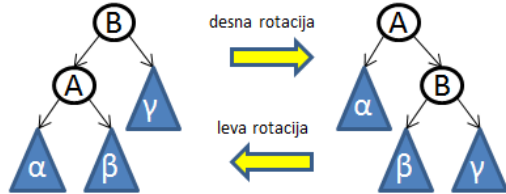
VSTAVLJANJE V KOREN BST



Vstavite v koren BST: 8, 5, 19, 12 in 17.

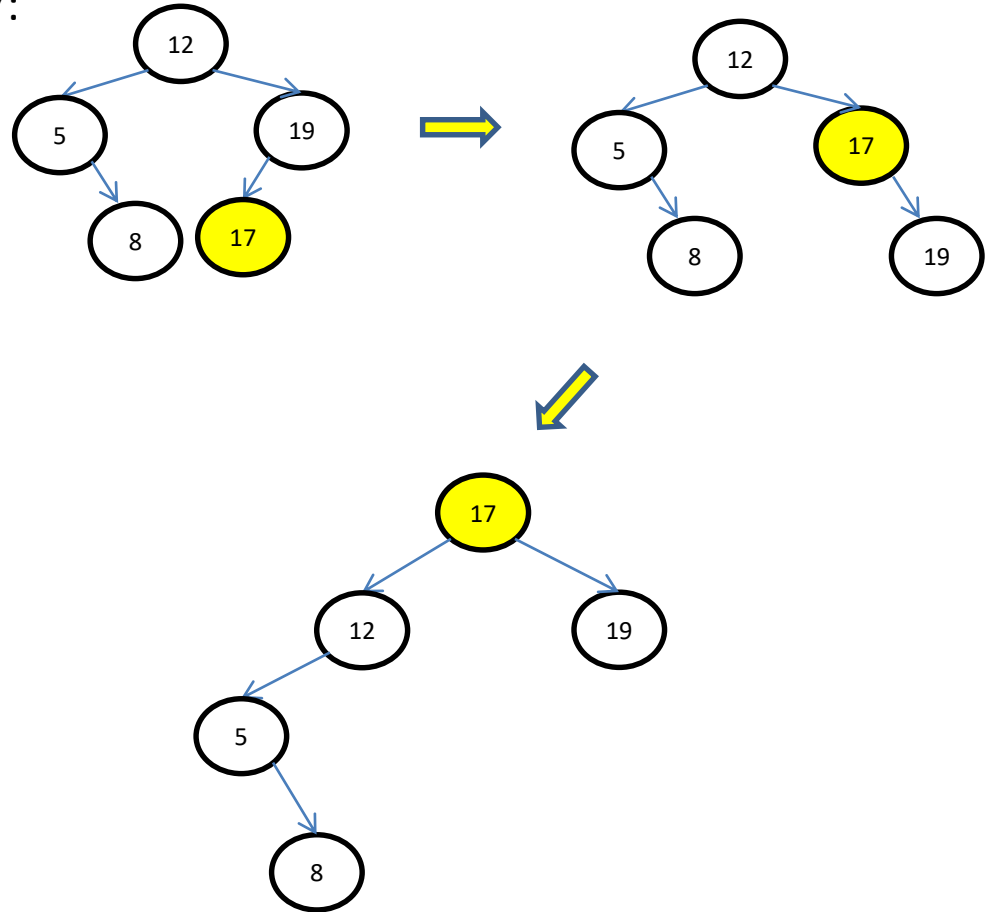


VSTAVLJANJE V KOREN BST



Vstavite v koren BST: 8, 5, 19, 12 in 17.

17:

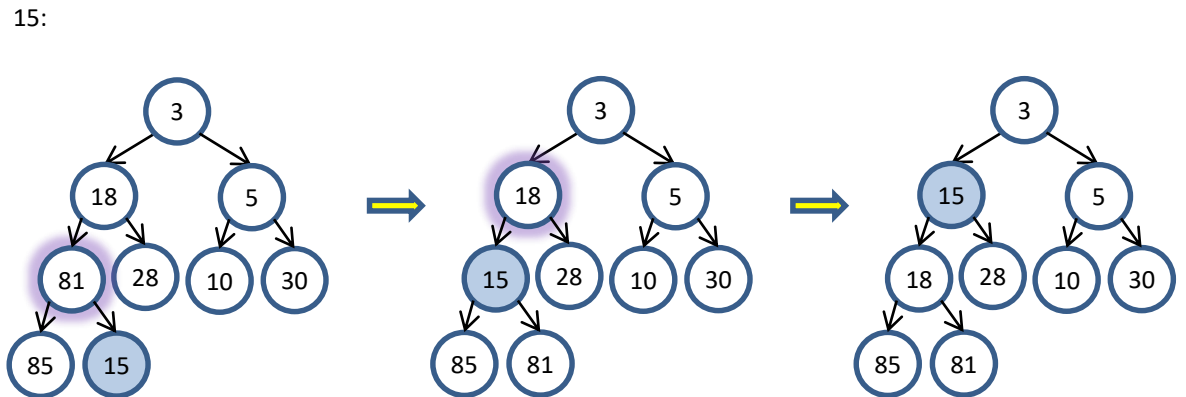
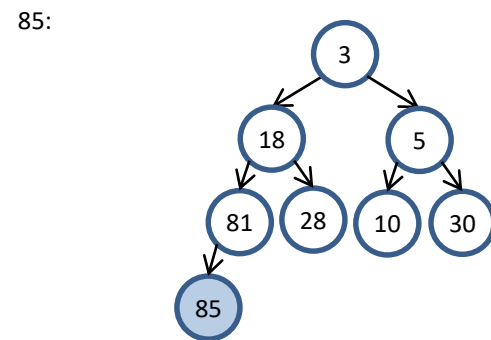
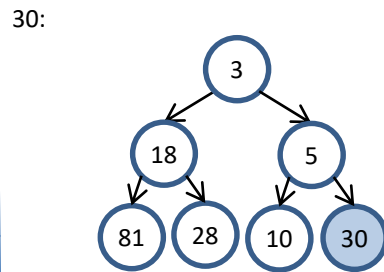
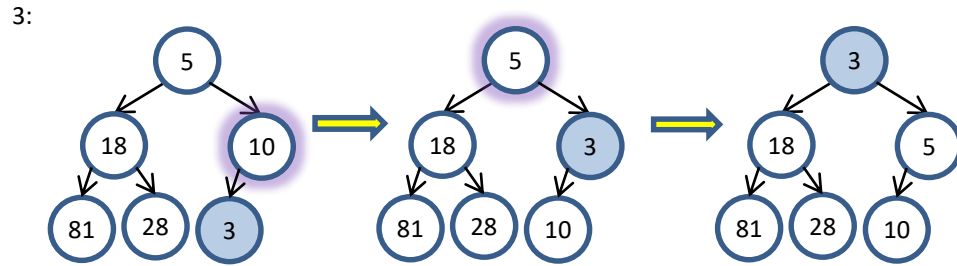
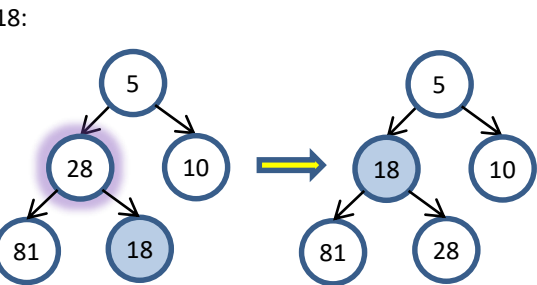
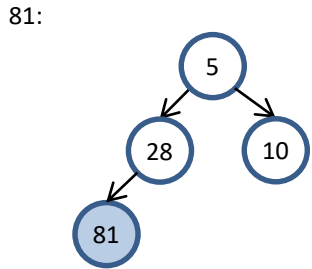
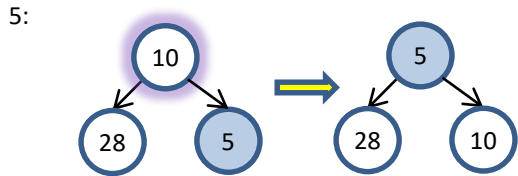
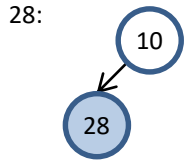


Realizacija operacij na kopici

Dodajanje elementa v kopico:

- element x najprej dodamo na prvo prazno mesto z leve na zadnjem nivoju drevesa
- če je zadnji nivo zapolnjen, ga dodamo kot prvega z leve na naslednjem nivoju
- zamenjujemo x z očetom, dokler ni izpolnjeno eno izmed:
 - oče manjši od x
 - x v korenu drevesa

Vstavite v kopico: 10, 28, 5, 81, 18, 3, 30, 85, 15



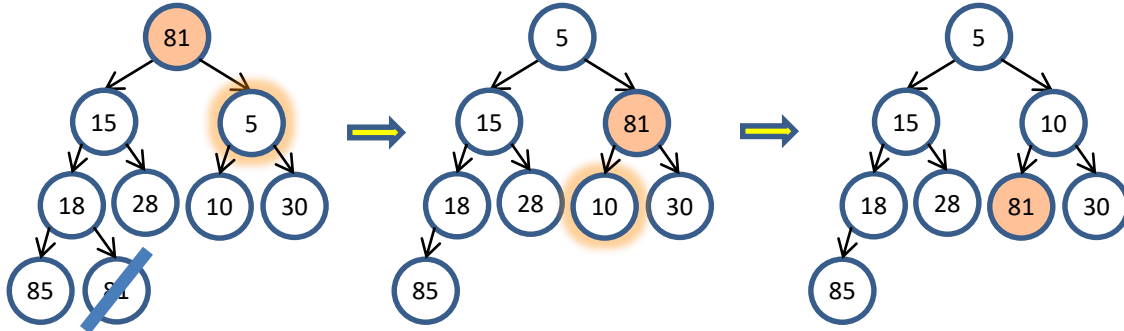
Realizacija operacij na kopici

Brisanje najmanjšega elementa:

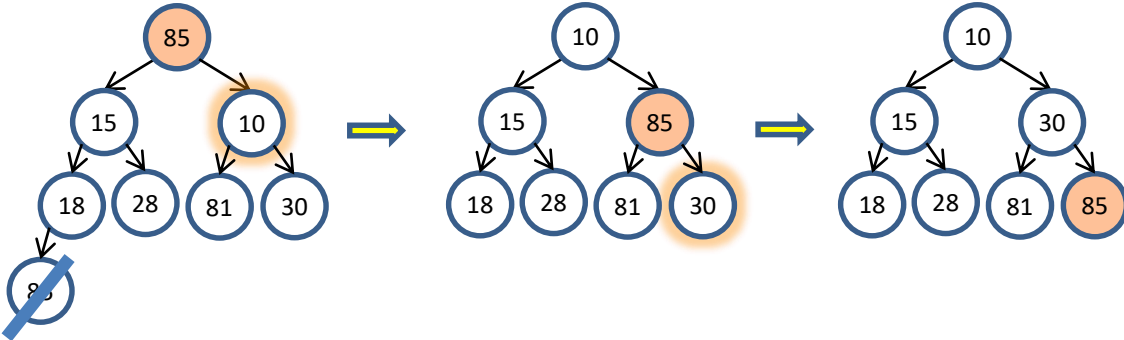
- je v korenu
- nadomestimo ga z najbolj desnim elementom x na zadnjem nivoju kopice
- zaporedno zamenjujemo x z manjšim od obeh sinov, dokler ni izpolnjeno vsaj eno:
 - x manjši od obeh sinov
 - x list drevesa

Trikrat odstranite najmanjši element:

3



5



10

