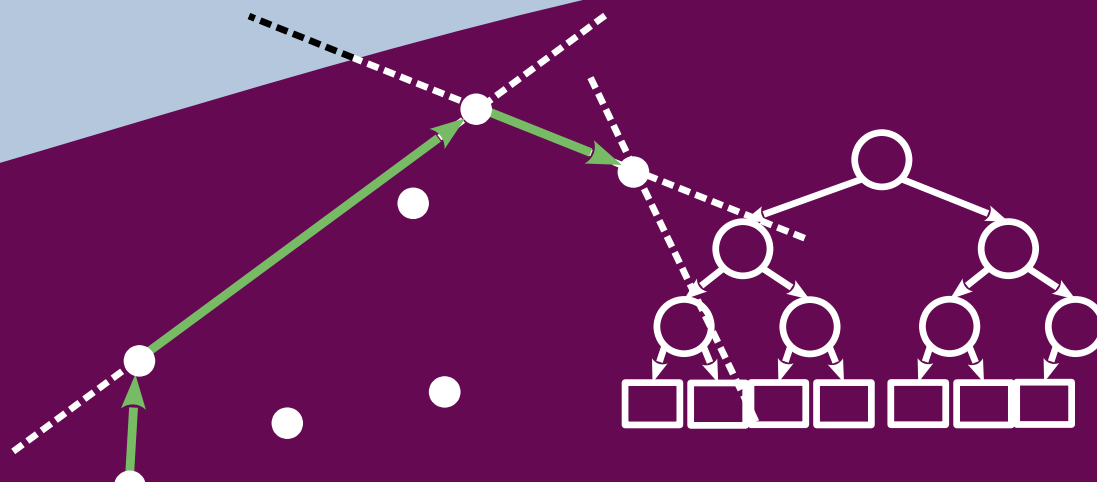
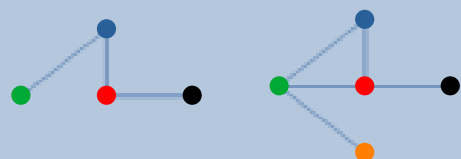


# Napredni algoritmi i strukture podataka

predavanja

2021./2022.

## Uravnotežena stabla (*Balanced Trees*)



# Creative Commons



- slobodno smijete:

- dijeliti — umnožavati, distribuirati i javnosti priopćavati djelo
- prerađivati djelo



- pod sljedećim uvjetima:

- imenovanje: morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- nekomercijalno: ovo djelo ne smijete koristiti u komercijalne svrhe.
- dijeli pod istim uvjetima: ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



*U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela.*

*Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.*

*Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.*

*Tekst licence preuzet je s <http://creativecommons.org/>*

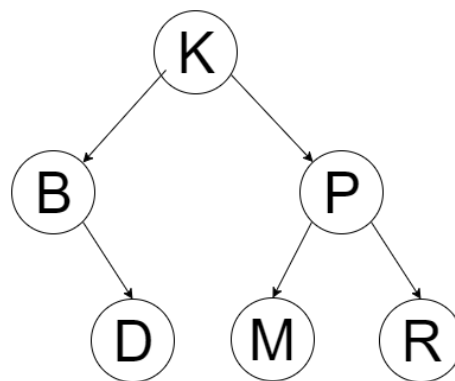
# Sadržaj

- Motivacija
- Uravnoteženo stablo
- Savršeno uravnoteženo stablo
- Uravnotežavanje stabla
- DSW algoritam
  - Rotacije
  - Koraci
- AVL stabla
  - Dodavanje
  - Brisanje
- Samopodešavajuća stabla

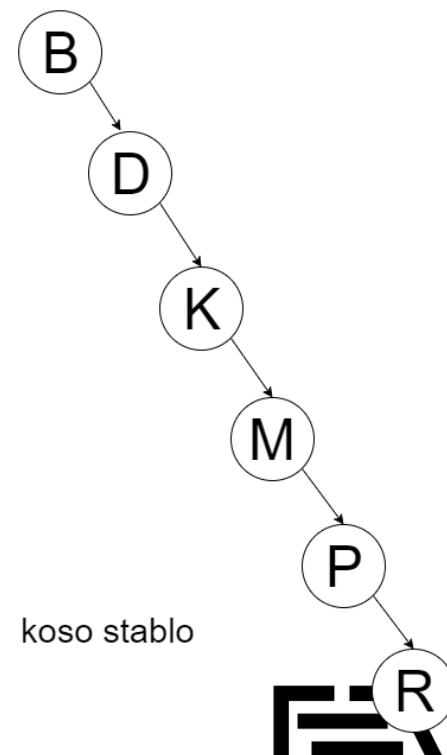
# Motivacija

- Liste – linearne, pretraživanje  $O(n)$
- Stabla – hijerarhijska rekurzivna struktura
- → stabla **prikladne** strukture –  $O(\log n)$  pretraživanje
- → stabla **neprikladne** strukture –  $O(n)$

- **Uravnotežavanje stabla!**



stablo

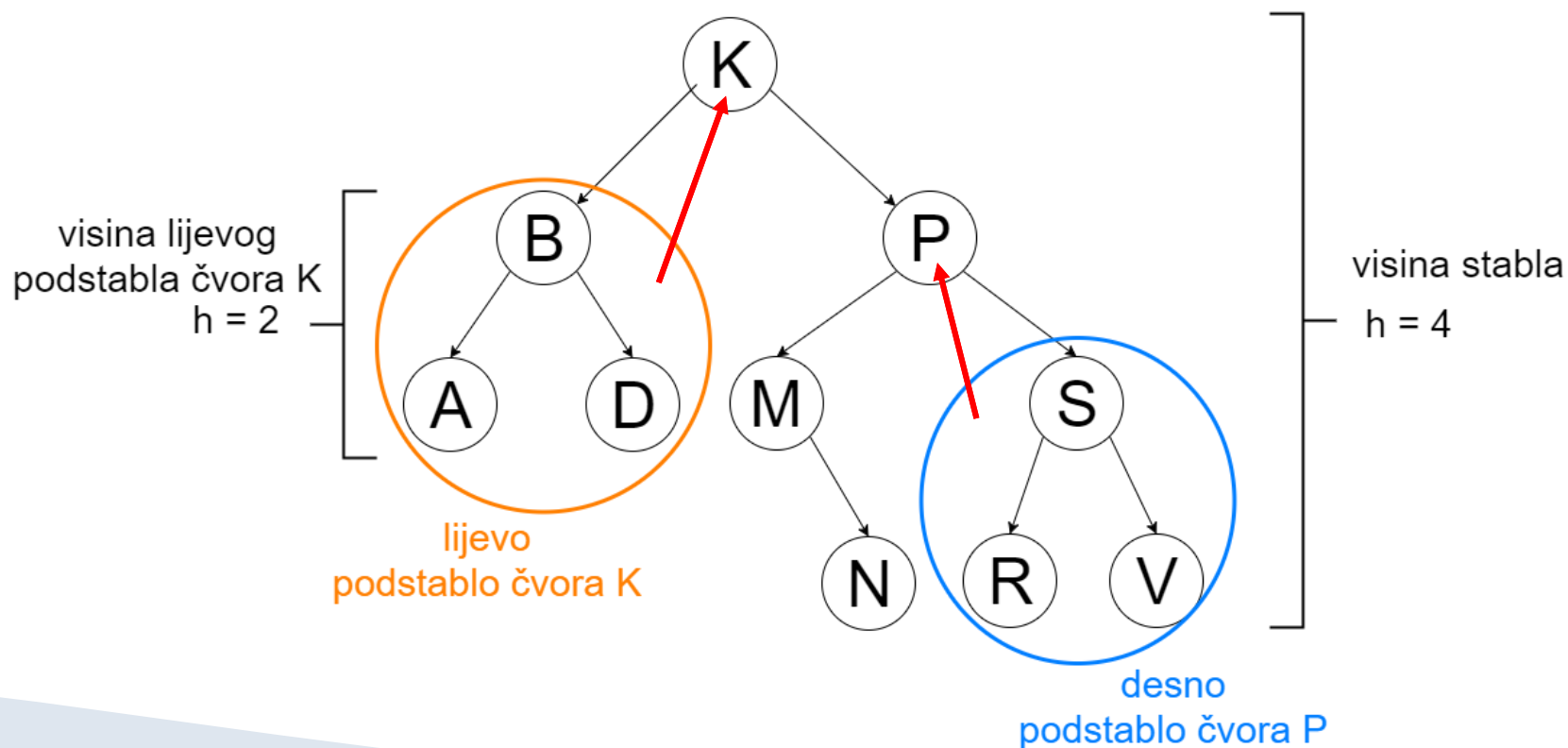


koso stablo



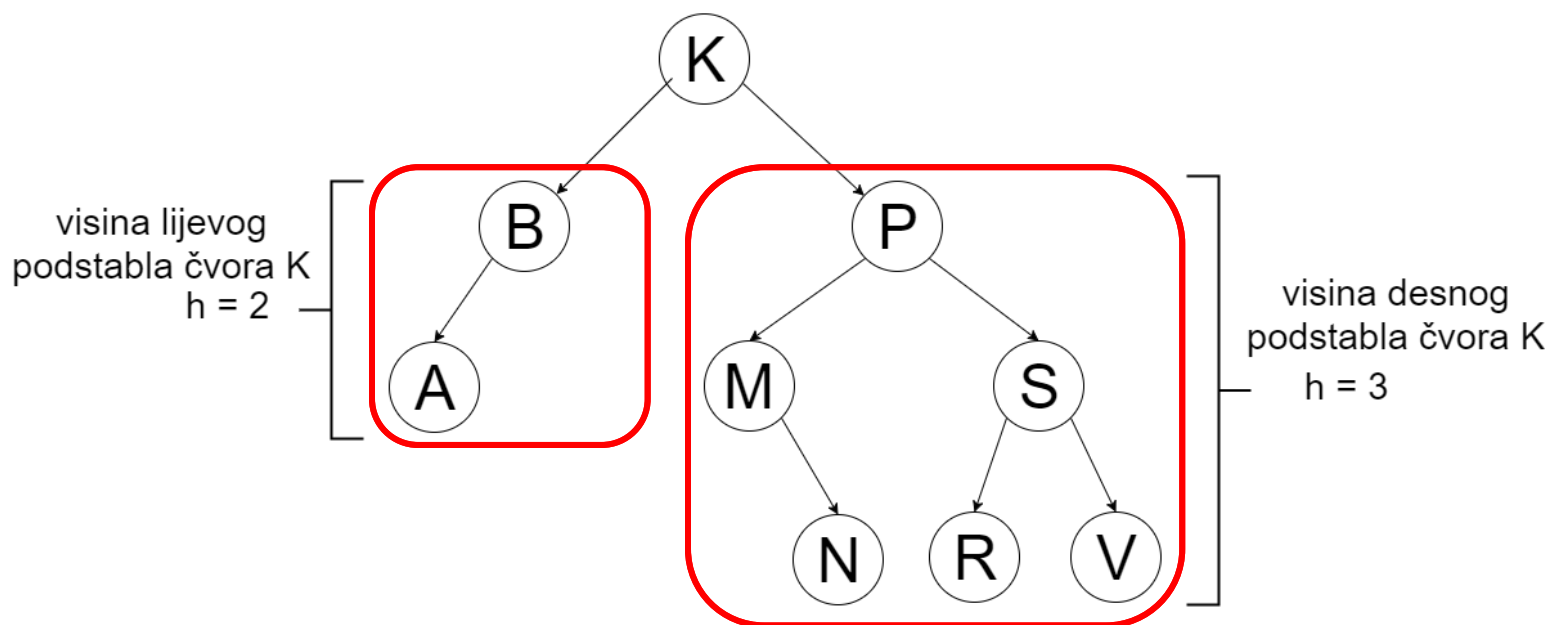
# Uravnoteženo stablo

- **Podstablo:** čvor i svi njegovi potomci
- **Visina stabla  $h$ :** Duljina najdužeg puta od korijena do lista



# Uravnoteženo stablo

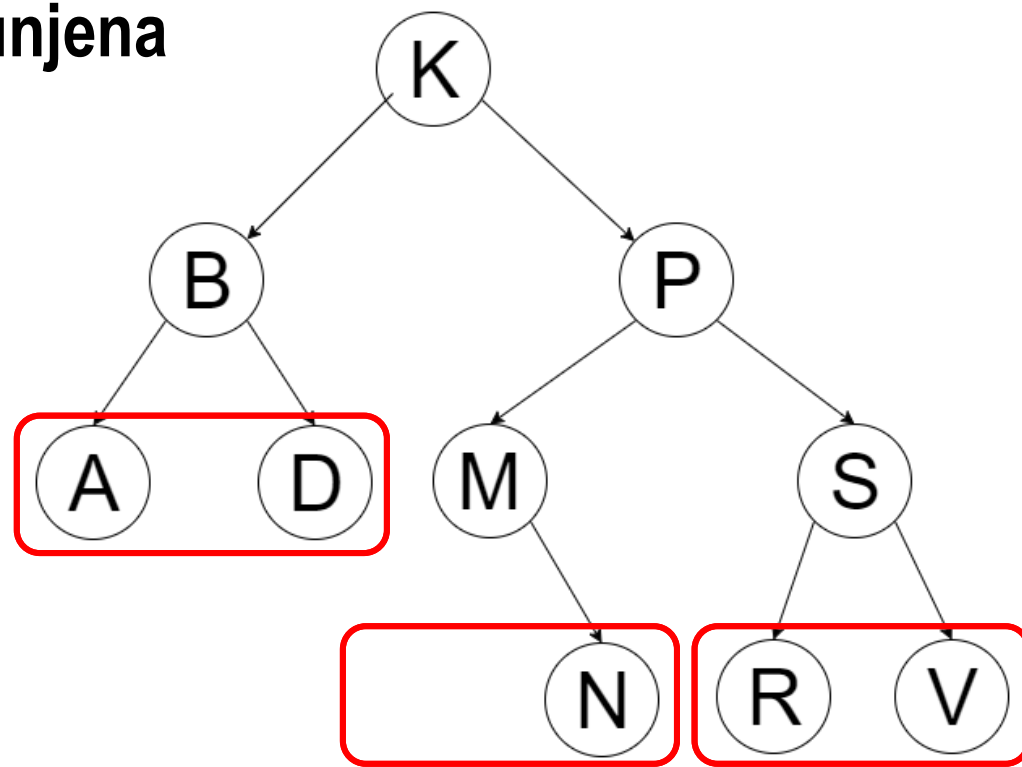
- Uravnoteženo stablo:** razlika visina podstabala svakog čvora najviše 1



- $3 - 2 = 1$
- $\rightarrow$  uvjet vrijedi za čvor K

# Savršeno uravnoteženo stablo

- Najpoželjnija struktura stabla po pitanju brzine pretraživanja
- **Definicija: Uravnoteženo stablo kojemu jedino najniža razine smije biti djelomično popunjena**
- Svi su listovi takvog stabla u najviše dvije razine



savršeno uravnoteženo stablo

# Sadržaj

- Motivacija
- Uravnoteženo stablo
- Savršeno uravnoteženo stablo
- **Uravnotežavanje stabla**
- DSW algoritam
  - Rotacije
  - Koraci
- AVL stabla
  - Dodavanje
  - Brisanje
- Samopodešavajuća stabla



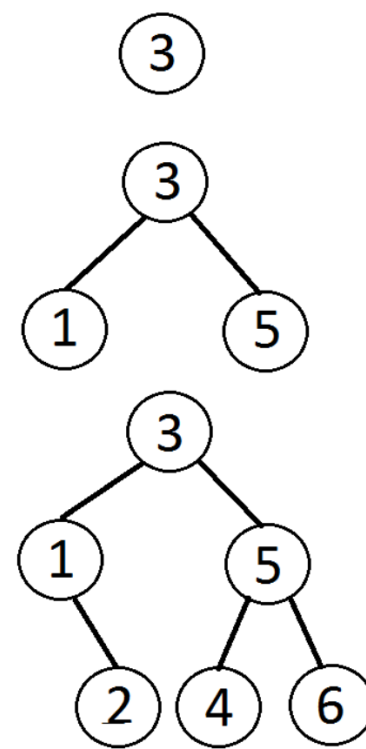
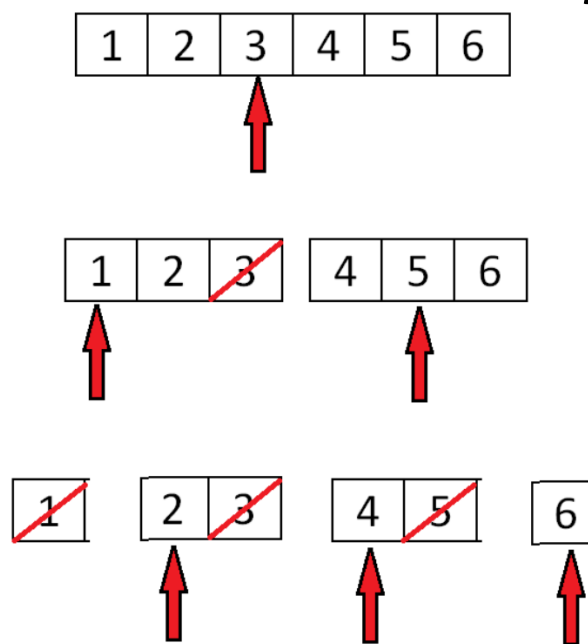
# Uravnotežavanje stabla

- Kako dobiti uravnoteženo stablo?
  - 1) Promišljenim redoslijedom upisa podataka
    - pripremimo podatke prije izgradnje stabla
  - 2) Restrukturiranjem stabla
    - stablo već postoji, pa ga uravnotežavamo različitim postupcima

# Uravnotežavanje stabla

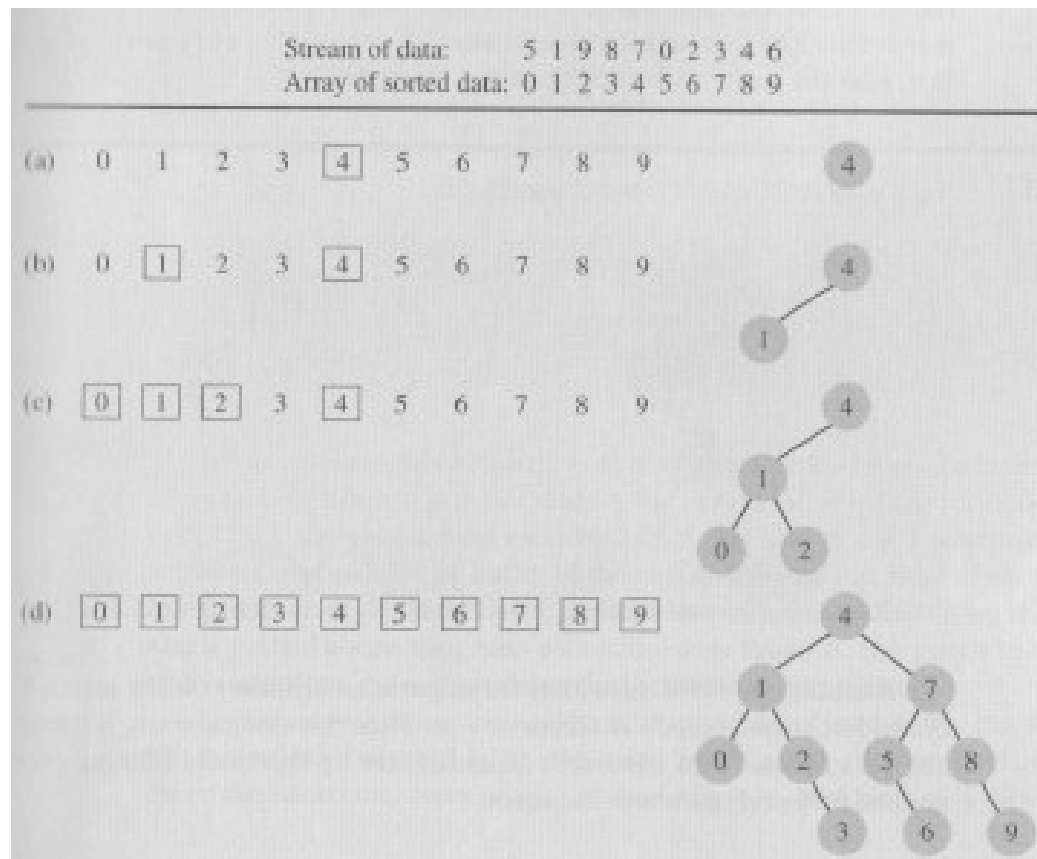
- Promišljenim redoslijedom upisa podataka:
  - Prikupiti** sve podatke koji će doći na ulaz, **sortirati** ih u niz
  - Pronaći **središnji element** niza – on postaje korijen
  - Središnji elementi lijevog i desnog podniza postaju njegova djeca
  - Rekurzivno ponavljamo korake 2-3 dok u stablo ne smjestimo sve elemente niza

→ Ovim postupkom uvijek dobijemo savršeno uravnoteženo binarno stablo



# Uravnotežavanje stabla

```
MakeBalTree (data[], left, right):  
    if left <= right:  
        middle = (left + right)/2  
        insert data[middle] into the tree  
        MakeBalTree (data, left, middle-1)  
        MakeBalTree (data, middle+1, right)
```



# Uravnotežavanje stabla

- **Nedostaci:**

1. Nemamo uvijek sve ulazne podatke odjednom – obično podaci stalno pristižu, a i iz postojećeg stabla se brišu i mijenjaju.
2. Podatke treba sortirati – to može biti zahtjevno – i memorijski i vremenski.
3. Potreba za dodatnom memorijom.

# Uravnotežavanje stabla

## 2) Restrukturiranjem stabla

- U praksi se stabla najčešće uravnotežavaju restrukturiranjem
- Postoje razni postupci - naučit ćemo neke od najpoznatijih
  - DSW algoritam
  - AVL stabla
  - RB-stabla
  - B-stabla
  - ...

# Sadržaj

- Motivacija
- Uravnoteženo stablo
- Savršeno uravnoteženo stablo
- Uravnotežavanje stabla
- **DSW algoritam**
  - Rotacije
  - Koraci
- AVL stabla
  - Dodavanje
  - Brisanje
- Samopodešavajuća stabla

# DSW algoritam - uvod

- DSW algoritam
  - Colin **Day** („Balancing a Binary Tree”, 1976.), nakon toga Quentin F. **Stout** i Bette L. **Warren**, „Tree Rebalancing in Optimal Time and Space ”, 1986.
- Daje **savršeno uravnoteženo** stablo.
- Temelji se na rotacijama.
- **Rotacija** – postupak kojim dijete postaje roditelj, a roditelj dijete pritom se moraju poštovati definicijska pravila stabla – ne smije se narušiti logika kojom se stablo gradilo, jer onda bi pretraživanje davalo krive rezultate.

# DSW algoritam – pojašnjenje rotacija

- Rotacija čvora C oko čvora B = desna rotacija

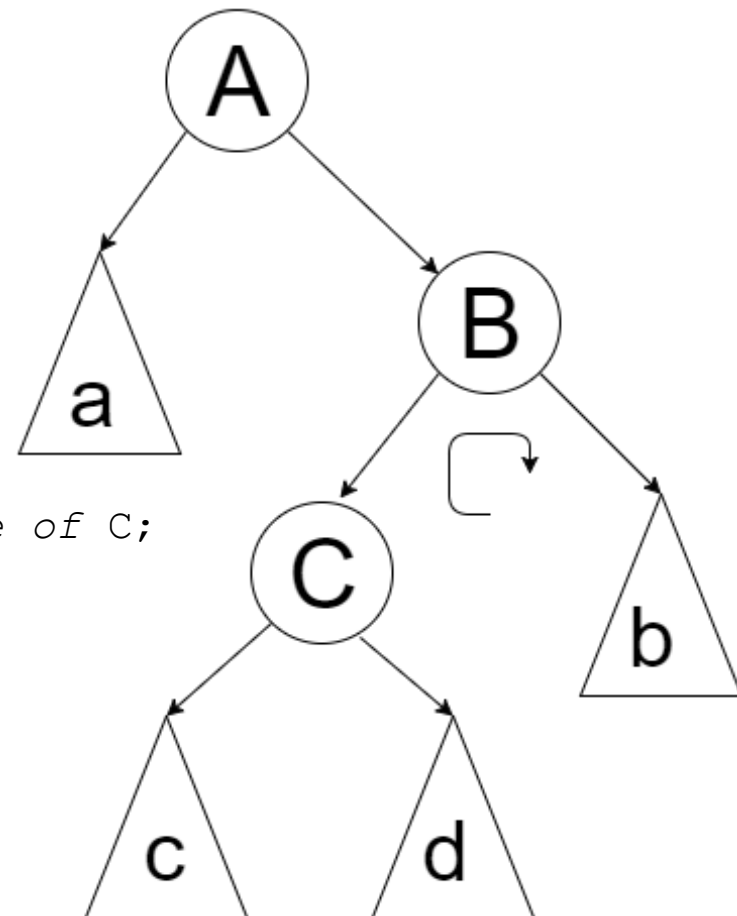
RightRotation (A, B, C):

if B is not root: // i.e., A not NULL

redirect pointer in A to C;

redirect left pointer of B to right subtree of C;

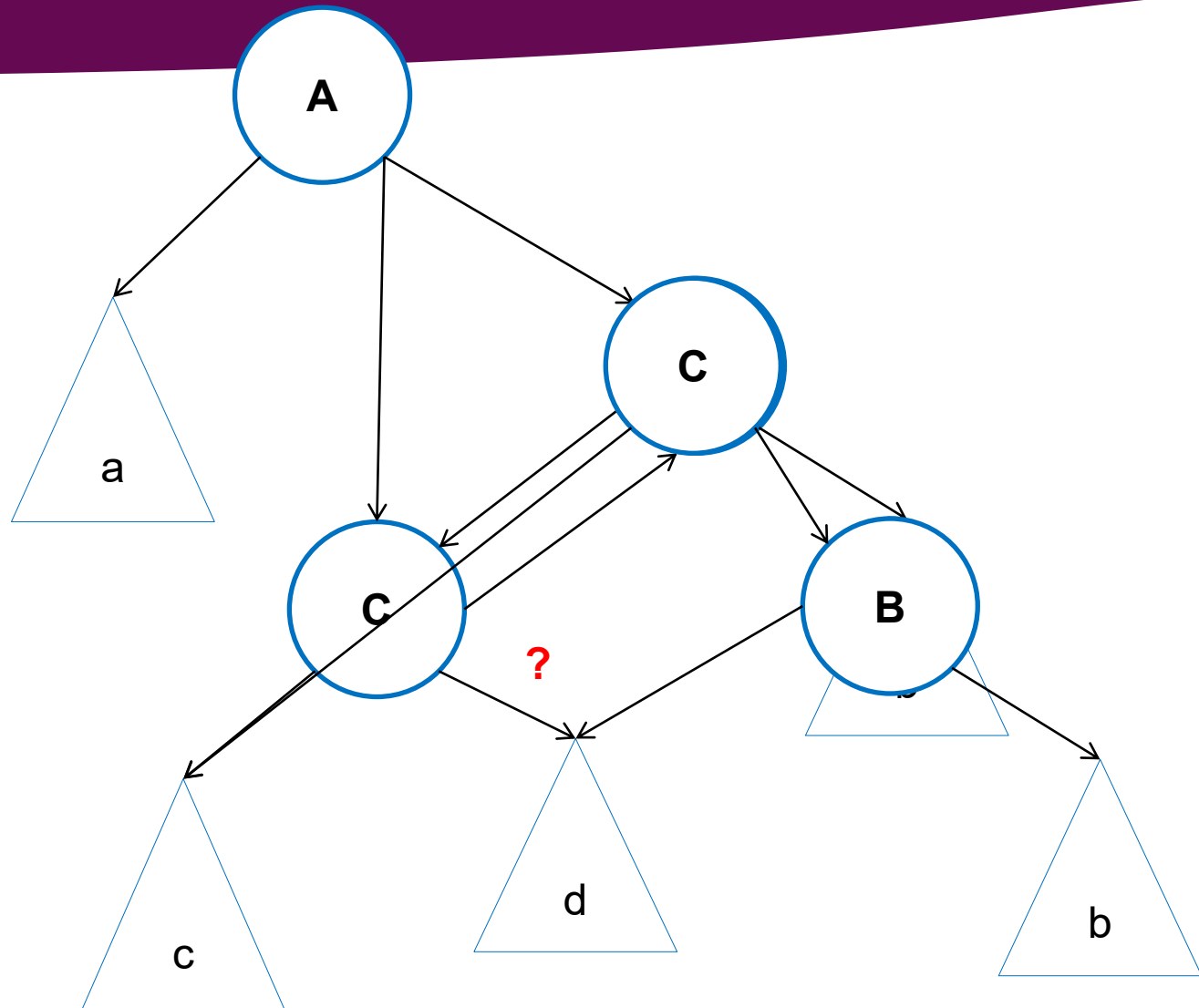
redirect right pointer of C to B;



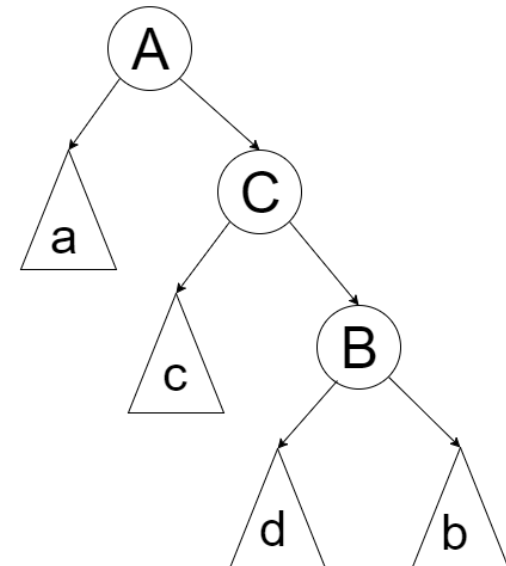
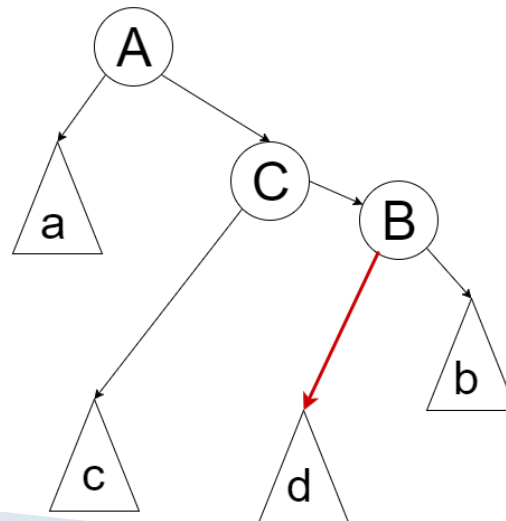
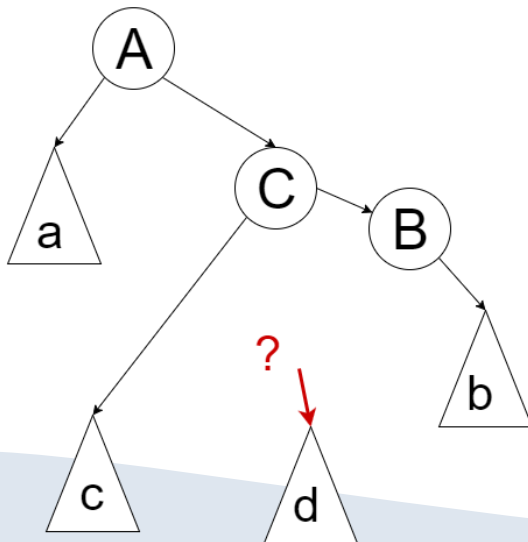
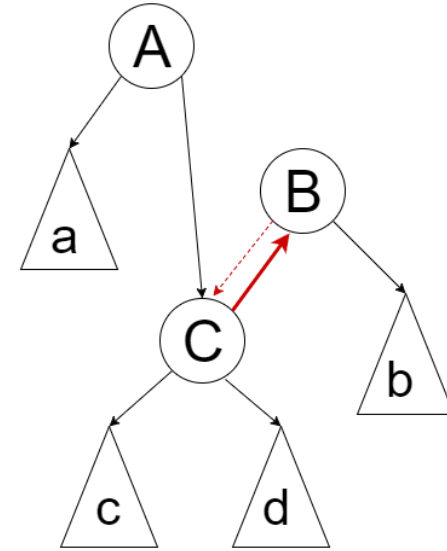
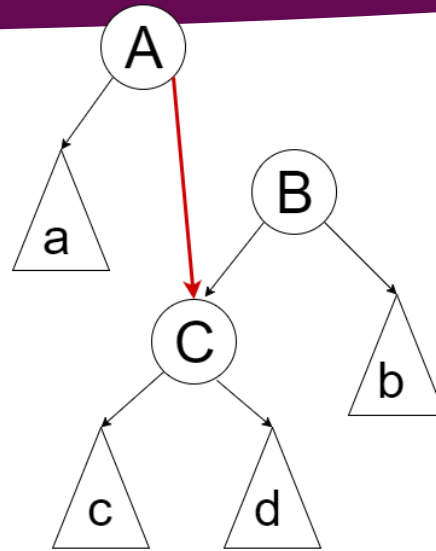
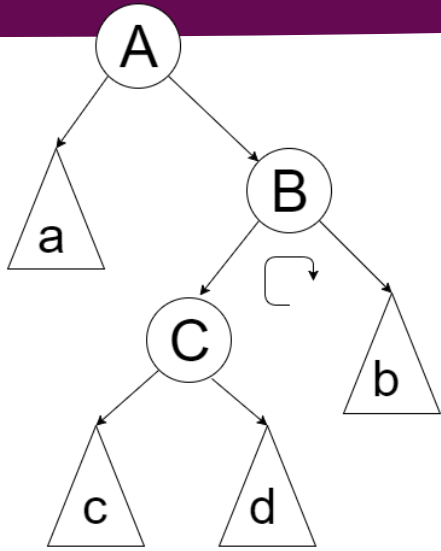
**Napomena:** za provjeru ispravnosti – nakon svake rotacije provjerite je li *inorder* ispis stabla ostao isti. Ako nije, rotacija je pogrešno provedena!



# Vizualizacija rotacije – desna rotacija – C oko B



# Vizualizacija rotacije – desna rotacija – C oko B

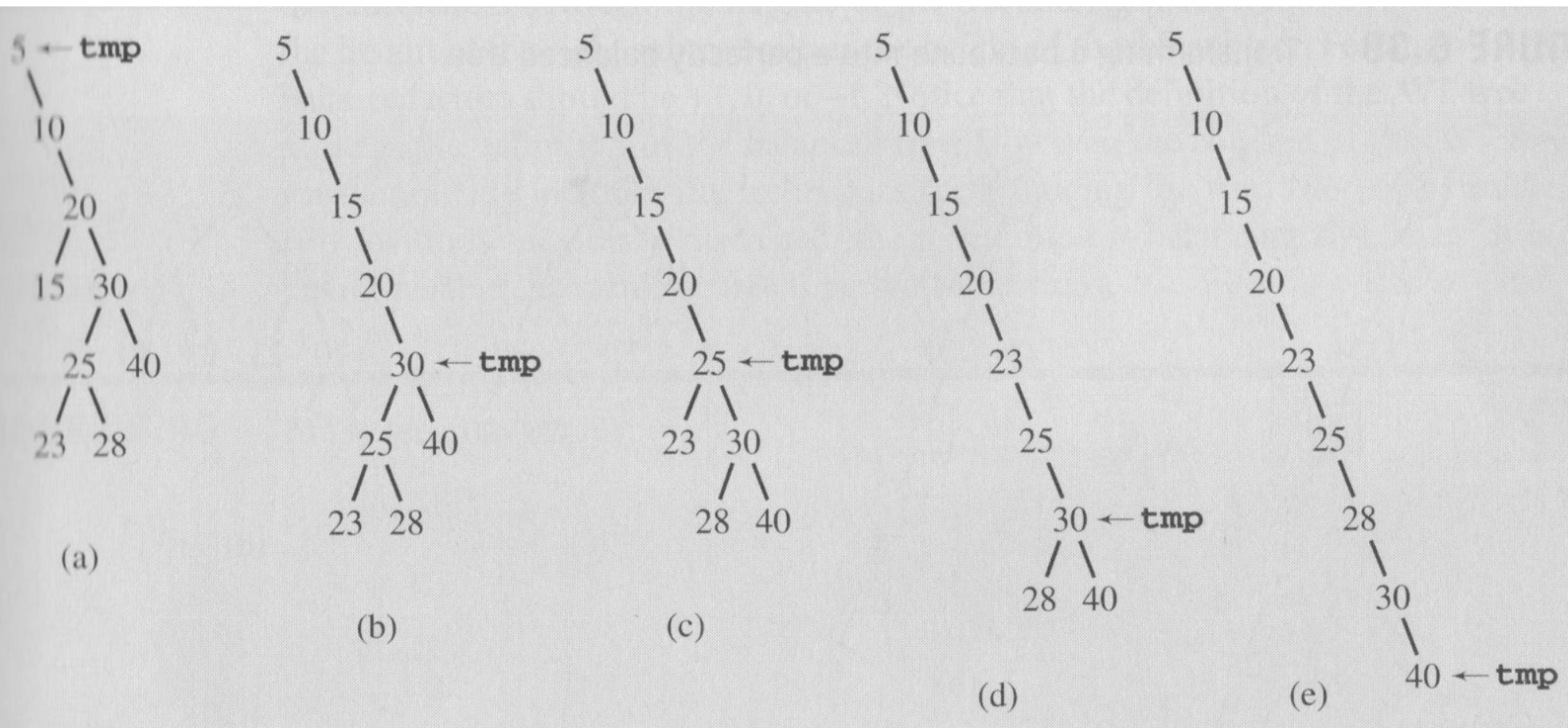


# DSW algoritam

- DSW algoritam ima dvije faze:
  - 1) Pretvaranje binarnog stabla u **koso stablo** – naziva se ***kralježnica*** (*backbone, vine*)
    - radimo rotacije u jednu stranu dok ne dobijemo koso stablo
    - složenost  **$O(n)$**
  - 2) Pretvaranje *kralježnice* u savršeno uravnoteženo stablo
    - radimo rotacije u suprotnu stranu dok ne dobijemo savršeno uravnoteženo stablo
    - složenost  **$O(n)$**
- → ukupna složenost:  **$O(n)$**

# DSW algoritam

- 1) Pretvaranje binarnog stabla u *kralježnicu* – rotacije npr. udesno



# DSW algoritam

## 1) Pretvaranje binarnog stabla u *kralježnicu* – rotacije npr. udesno

```
CreateBackbone (root):
```

```
    tmp = root
```

```
    while (tmp != NULL):
```

```
        if tmp has left child ch:
```

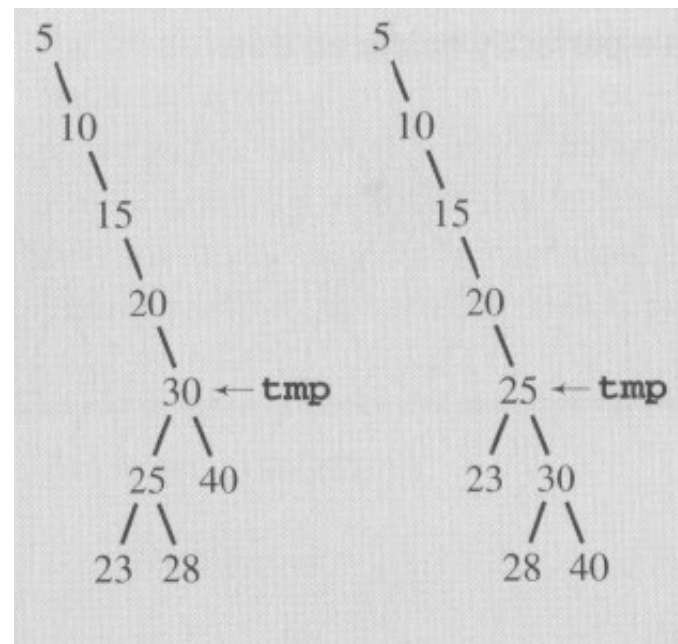
```
            right rotate ch about tmp
```

```
            redirect tmp to the ch
```

```
        else:
```

```
            redirect tmp to its right child;
```

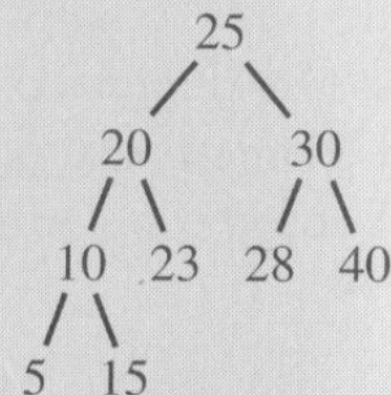
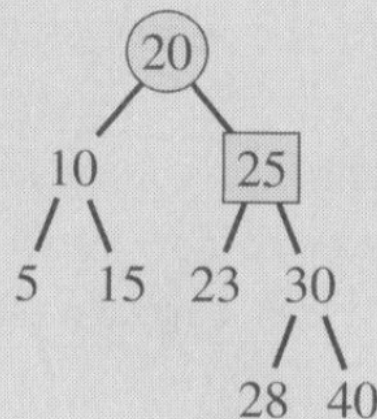
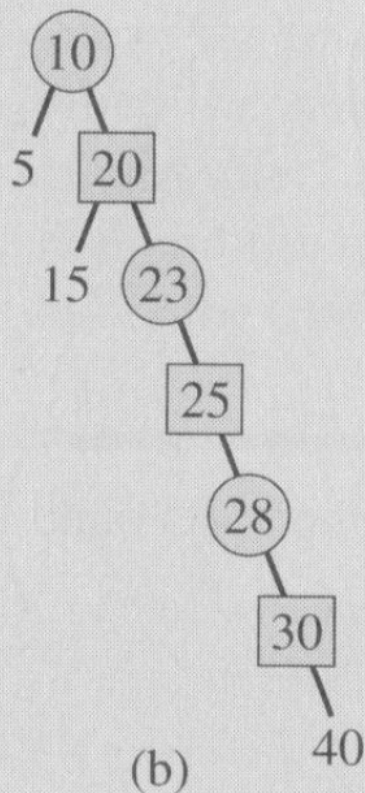
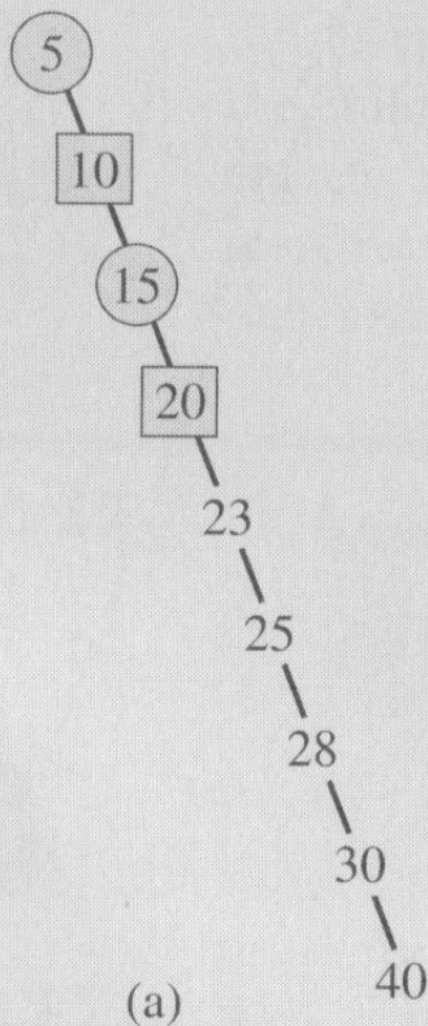
- Najgori slučaj: lijevo koso stablo
  - najviše  $(n-1)$  rotacija složenosti  $O(1)$
  - while uvjet se ispita  $2(n-1)+1$
  - Ukupno:  **$O(n)$**





# DSW algoritam

## 2) Pretvaranje *kralježnice* u savršeno uravnoteženo stablo



# DSW algoritam

## 2) Pretvaranje *kralježnice* u uravnoteženo stablo – rotacije npr. ulijevo

CreatePerfectTree (n):

$h = \lfloor \log_2(n+1) \rfloor$   
 $h \left\{ \begin{array}{l} k = 2^h - 1 \text{ // broj čvorova u punim razinama} \\ n-k \text{ left rotations of every second node from the top} \\ \text{while } (k > 1): \\ \quad k = k/2 \\ \quad k \text{ left rotations of every second node from the top;} \end{array} \right.$

- **n-k** je broj elemenata u zadnjoj, nepopunjenoj razini
  - prvih **n-k** rotacija se pobrine za njih
- rotacije u while petlji se brinu za popunjene razine

Složenost faze:  **$O(n)$**  [Drozdek]

Ukupna složenost DSW-a:  **$O(n)$**

# Problem s DSW

- Ravnoteža stabla obično je narušena samo lokalno
- DSW radi globalno uravnotežavanje cijelog stabla, iako je to nekad možda nepotrebno
  - Složenost  $O(n)$
  - Možemo li bolje? **DA**
  - Lokalno uravnotežavanje
  - Cijena: uravnotežavanje ne mora biti savršeno



# Sadržaj

- Motivacija
- Uravnoteženo stablo
- Savršeno uravnoteženo stablo
- Uravnotežavanje stabla
- DSW algoritam
  - Rotacije
  - Koraci
- AVL stabla
  - Dodavanje
  - Brisanje
- Samopodešavajuća stabla

# AVL stabla

- AVL stabla (izvorni naziv *admissible tree*)
  - Georgii Maksimovič **Adelson Velskii**, Yevgenii Mikhailovič **Landis**, "An algorithm for the organization of information,,, 1962.
- AVL algoritam **uravnotežuje** stablo samo lokalno, no ne jamči *savršenu uravnoteženost* cijelog stabla
- **Definicija:** AVL stablo je stablo koje zadovoljava AVL definicijsko pravilo (opisano niže)
- *engl. balance factor*
- **Faktor ravnoteže FR** = (h desnog podstabla) – (h lijevog podstabla)
- **AVL definicijsko pravilo:** FR svakog čvora mora biti -1, 0 ili 1
- **Interpretacija:** Visine podstabala nekog čvora smiju se razlikovati samo za 1  $\Leftrightarrow$  definicija uravnoteženog stabla!

# AVL stabla

- Broj čvorova u h razina ne može biti manji od nekog minimuma, a iz definicijskog pravila slijedi rekurzivna relacija za minimum:

$$AVL_h = AVL_{h-1} + AVL_{h-2} + 1,$$

gdje su  $AVL_0 = 0$ ,  $AVL_1 = 1$  polazne vrijednosti

- Teorijske granice visine AVL stabla [Drozdek: Appendix A.5]

$$\log_2(n+1) \leq h \leq 1,44 \cdot \log_2(n+2) - 0,328$$

- Pretraživanje AVL stabla:  **$O(\log_2 n)$** 
  - u najgorem slučaju 44% sporije od pretraživanja savršenog stabla

# Dodavanje čvora u AVL stablo

Koraci za dodavanje čvora D u AVL stablo:

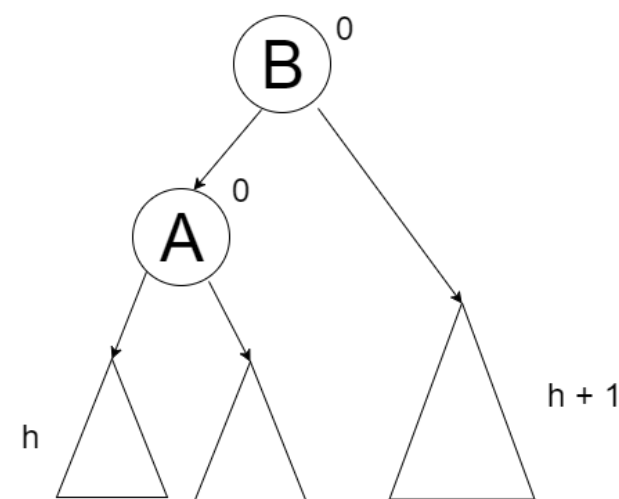
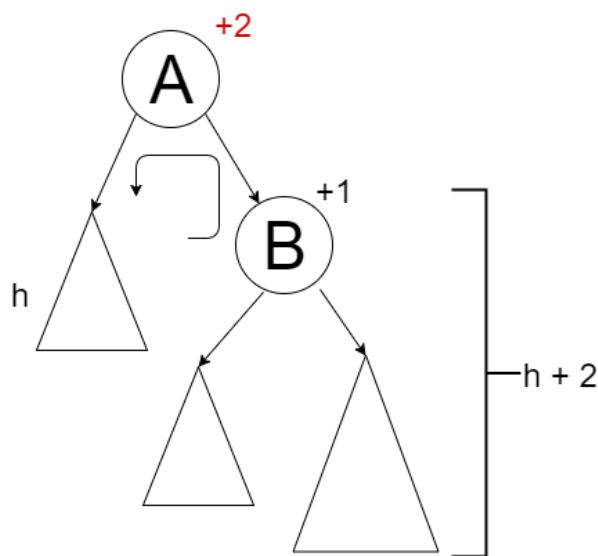
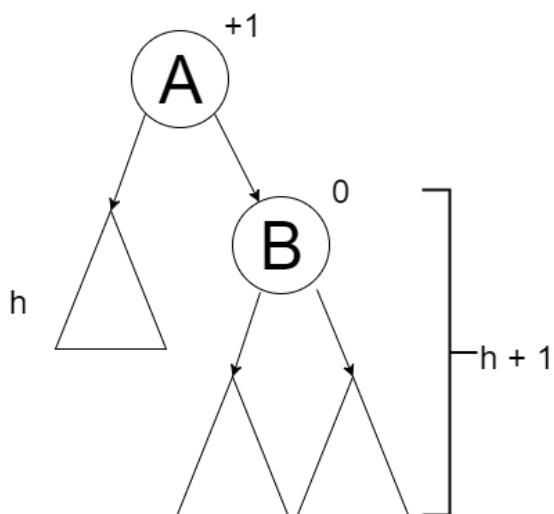
1. Pronaći mjesto gdje treba ići; dodati ga u stablo
2. Uravnotežiti stablo
  - Ažurirati faktor ravnoteže (FR) svih čvorova A uspinjući se od dodanog čvora D do korijena
  - Ako ažurirani  $FR(A)=0$ , gotovo uravnotežavanje
  - Ako ažurirani  $FR(A)=\pm 1$ , nastavi s ažuriranjem
  - Jedini koji zahtijeva intervenciju je prvi za kojeg je ažurirani  **$FR(A)=\pm 2$**

# Dodavanje čvora u AVL stablo

- Ako pri ažuriranju faktora ravnoteže čvora A dobijemo  $FR(A) = +/-2$ , potrebno je stablo ponovno **uravnotežiti**
- Uvedimo oznake  $X \in \{\text{desno}, \text{lijevo}\}$  i njegov komplement  $\neg X$  (onaj drugi)
- Moguća su dva opća slučaja:
  - 1) **izravnati**: čvor D dodan je u **X** podstablo **X** djeteta B čvoru A
    - **Jednaki predznaci**  $FR(A)$  i  $FR(B)$ 
      - $FR(A)=2$   $FR(B)=1$  ili  $FR(A)=-2$ ,  $FR(B)=-1$
  - 2) **izlomljeni**: čvor D dodan je u  **$\neg X$**  podstablo **X** djeteta B čvoru A
    - **Različiti predznaci**  $FR(A)$  i  $FR(B)$ 
      - $FR(A)=-2$   $FR(B)=1$  ili  $FR(A)=2$ ,  $FR(B)=-1$

# Dodavanje čvora u AVL stablo

- 1) **Izravnati:** Čvor D dodan je u **X** podstablo **X** djeteta B čvoru A
- Rješenje:
    - napravi rotaciju čvora B oko čvora A

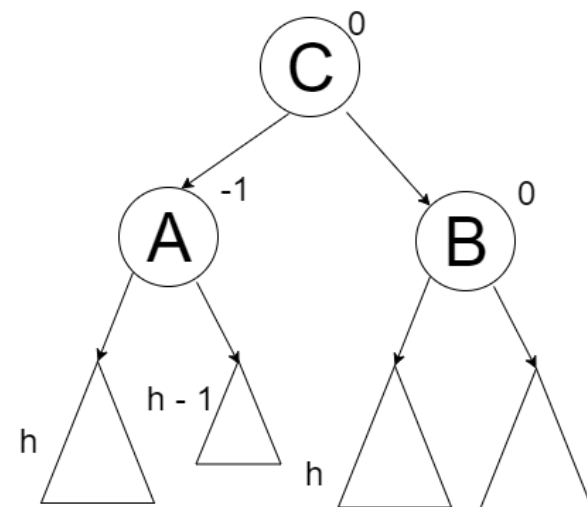
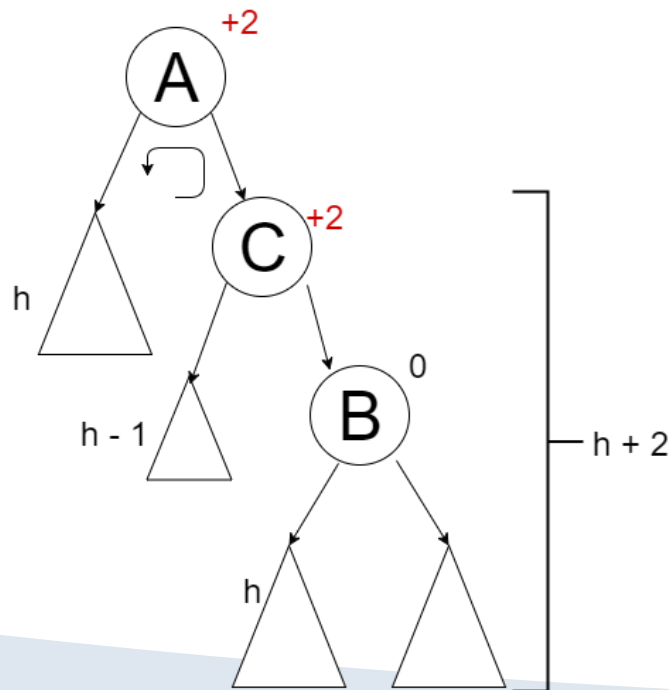
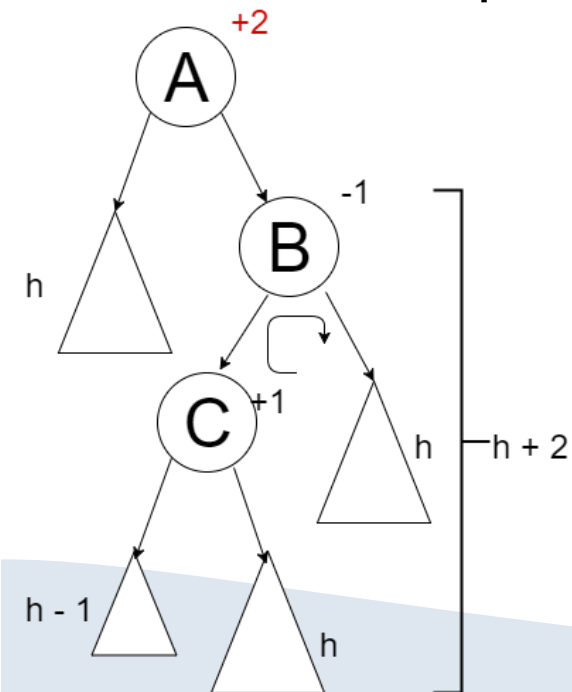


# Dodavanje čvora u AVL stablo

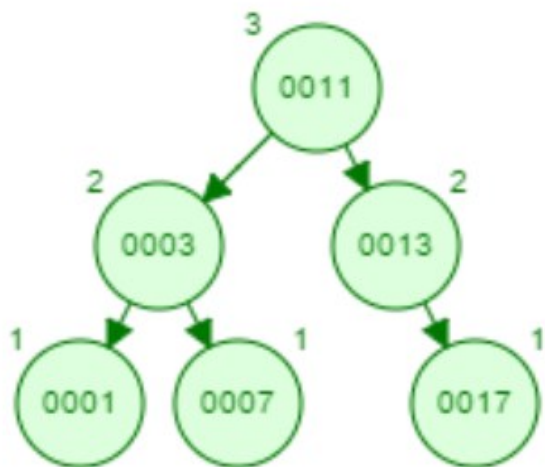
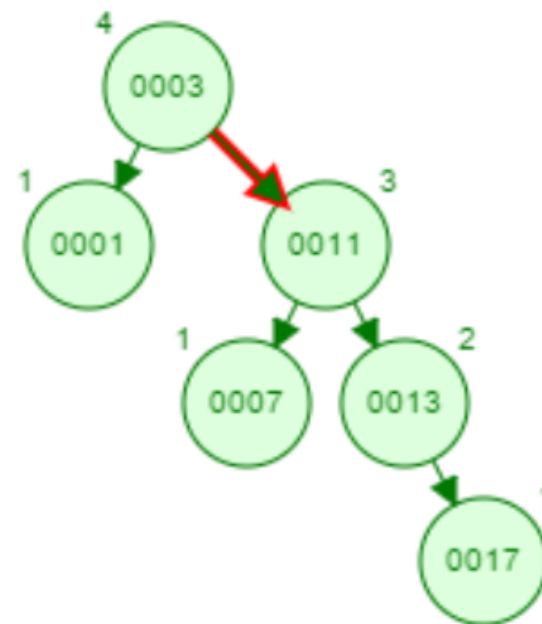
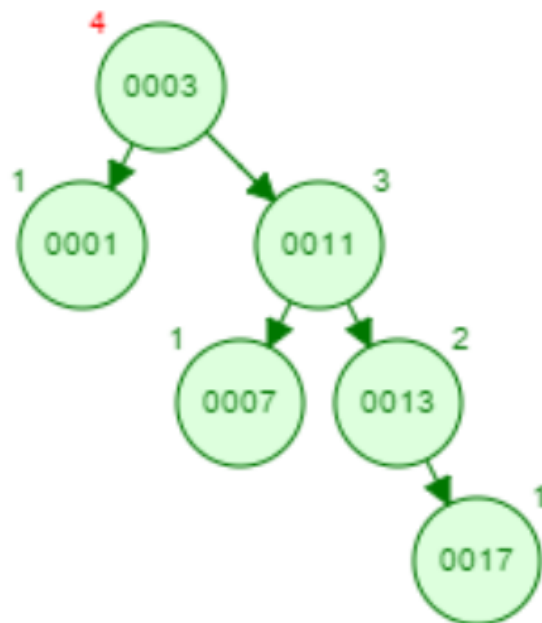
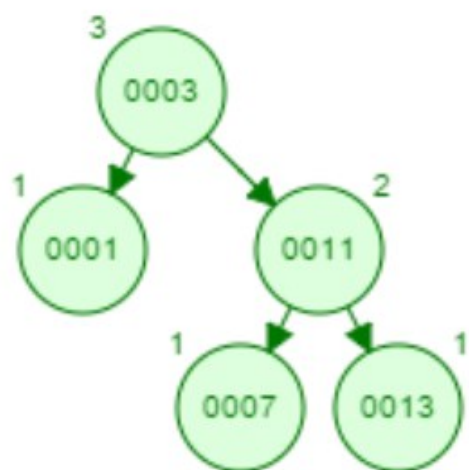
2) **Izlomljeni**: Čvor D dodan je u  $\neg X$  podstablo  $X$  djeteta B čvora A

- Rješenje:

- napravi rotaciju čvora C oko čvora B (pretvorba u **izravnati** oblik)
- napravi rotaciju čvora C oko čvora A



# Dodavanje čvora u AVL stablo



- primjer: dodavanje 17 u stablo



# Brisanje čvora u AVL stablu

Koraci za brisanje čvora D iz AVL stabla:

1. Ukloniti čvor **metodom *Deletion by Copying***
2. Uravnotežiti stablo (ne samo do prvog neuravnoteženog)
  - Ažurirati faktor ravnoteže (FR) svih čvorova A od mjesta s kojeg je uklonjen zamjenski čvor D' do korijena

## ***Mogućnosti pri ažuriranju***

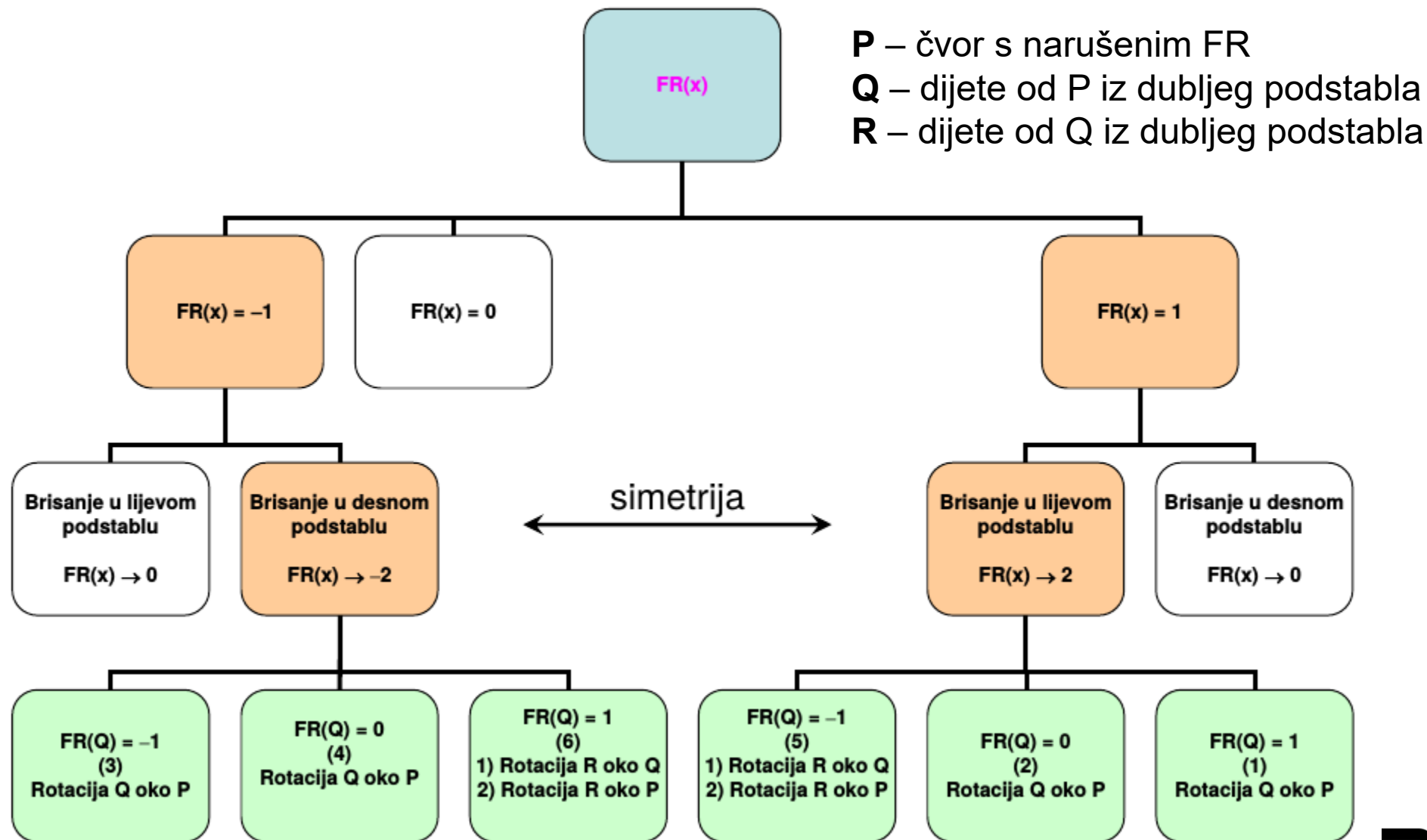
- FR(A) postao  $\pm 1 \rightarrow$  **KRAJ**
  - brisanje nije promijenilo visinu ovog podstabla
- FR(A) postao 0  $\rightarrow$  **NASTAVI**
  - brisanje je promijenilo visinu ovog podstabla
- FR(A) postao  $\pm 2 \rightarrow$  **INTERVENCIJA** (3 opća slučaja)
  - ako je nakon intervencije FR(A) =  $\pm 1$ ; gotovo uravnotežavanje, inače se nastavlja

# Brisanje čvora u AVL stablu

Postupak pri intervenciji:

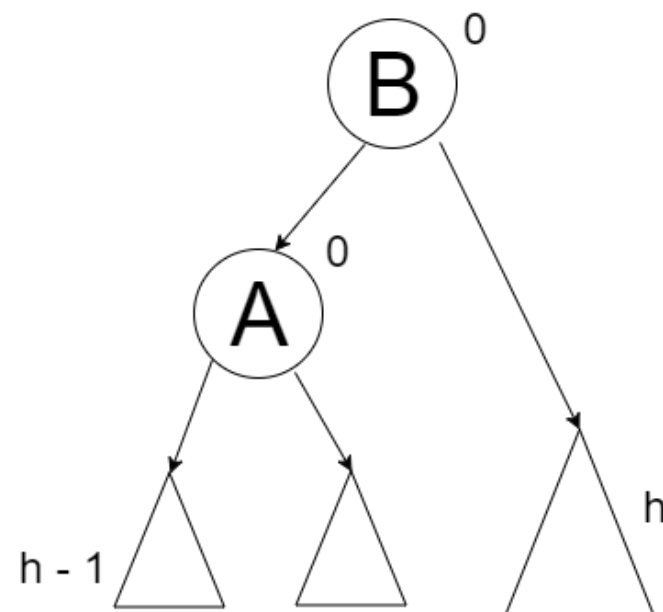
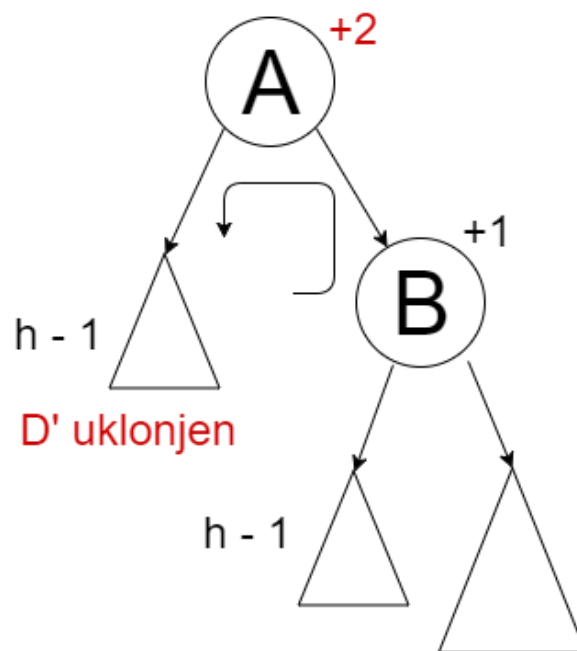
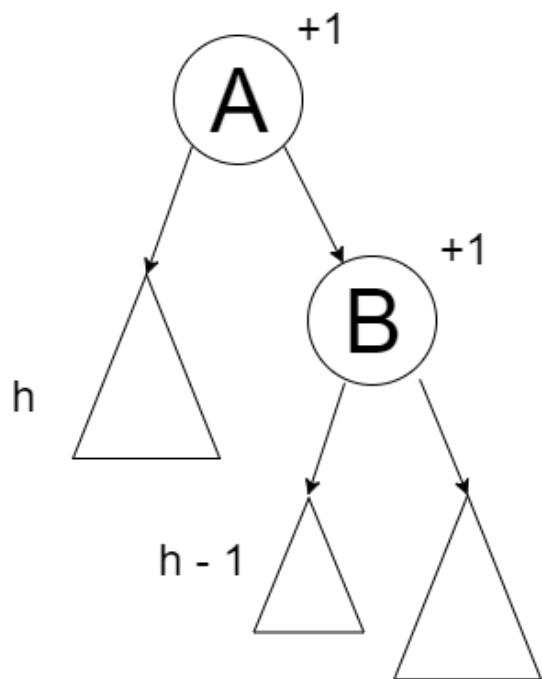
- Roditelj (čvor A) ima  $FR(A) = \pm 1$ , čvor je uklonjen iz „kraćeg” podstabla pa će  $FR(A)$  postati  $\pm 2$ , a njegovo dijete B u podstablu koje se nije mijenjalo može imati:
  - 1)  $FR(B) = -1$
  - 2)  $FR(B) = 0$
  - 3)  $FR(B) = +1$
- → 3 različita opća slučaja
  - 6 specifičnih slučajeva

# Brisanje čvora u AVL stablu – stablo odlučivanja



# Brisanje čvora u AVL stablu – koraci pri intervenciji

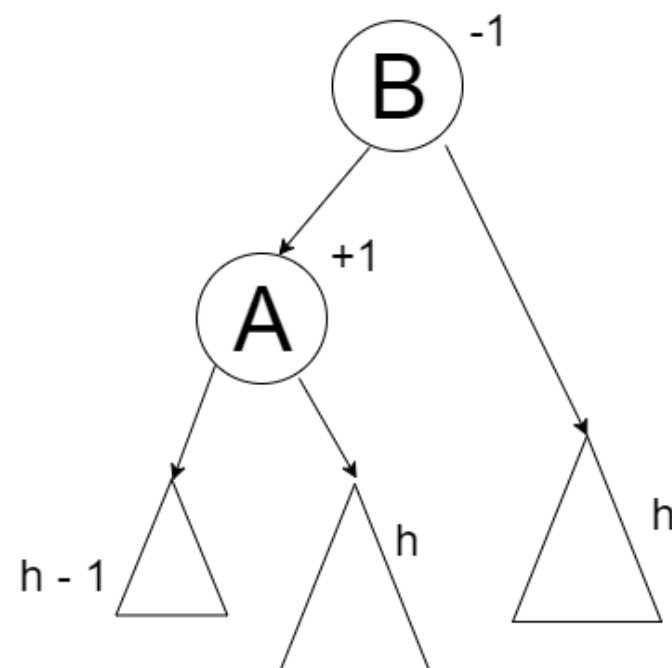
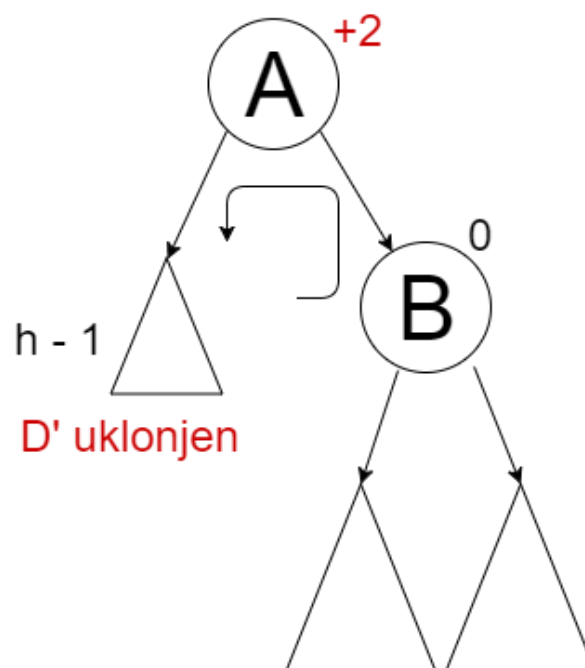
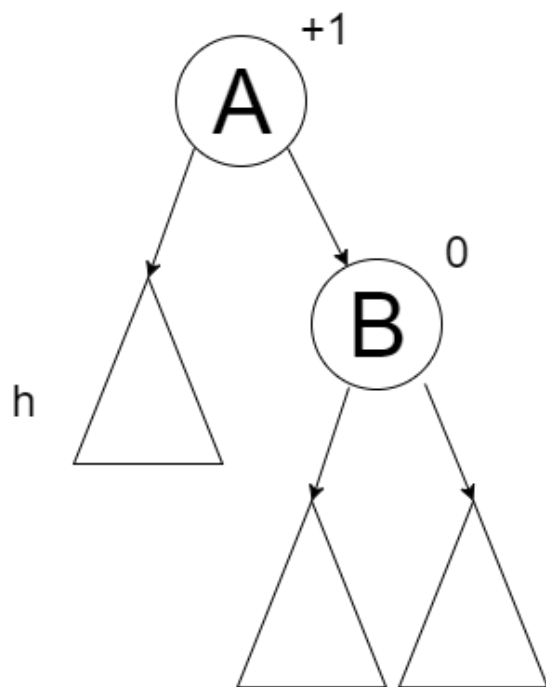
- 1)  $FR(A) = +2$  i  $FR(B) = +1$   $\leftrightarrow$   $FR(A) = -2$  i  $FR(B) = -1 \rightarrow$  identični predznaci
- Rješenje:** rotacija B oko A



# Brisanje čvora u AVL stablu – koraci pri intervenciji

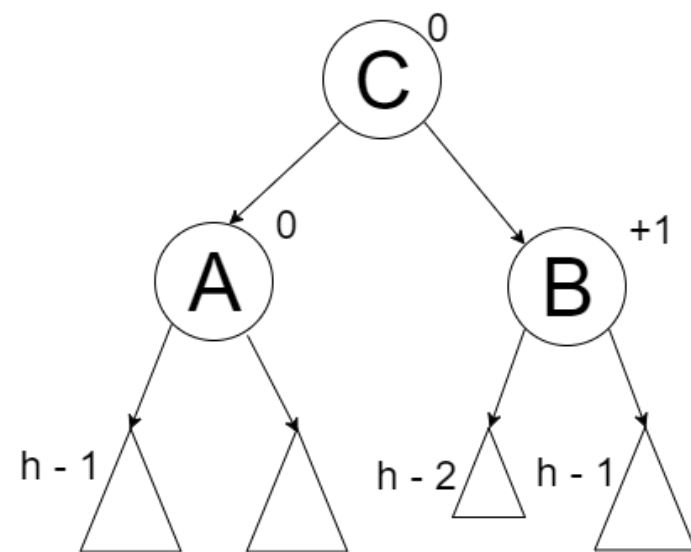
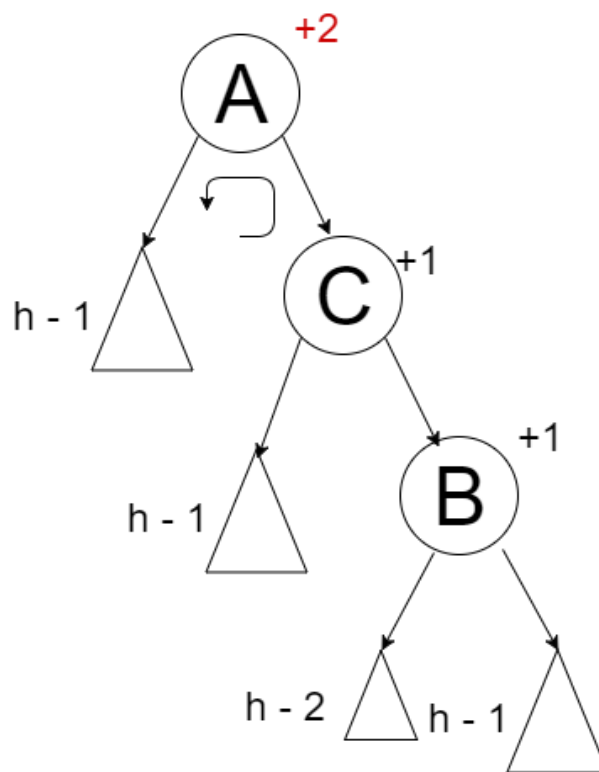
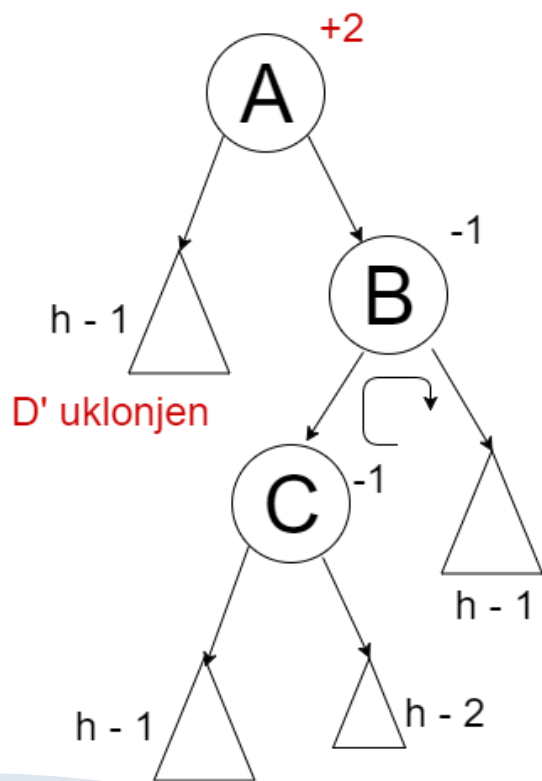
2)  $FR(A)=\pm 2$  i  $FR(B) = 0$

- **Rješenje:** rotacija B oko A (isto kao u 1. slučaju)



# Brisanje čvora u AVL stablu – koraci pri intervenciji

- 3)  $FR(A) = +2$  i  $FR(B) = -1$   $\leftrightarrow$   $FR(A) = -2$  i  $FR(B) = +1 \rightarrow$  suprotni predznaci
- Rješenje:** rotacija C oko B, pa rotacija C oko A



# Sadržaj

- Motivacija
- Uravnoteženo stablo
- Savršeno uravnoteženo stablo
- Uravnotežavanje stabla
- DSW algoritam
  - Rotacije
  - Koraci
- AVL stabla
  - Dodavanje
  - Brisanje
- Samopodešavajuća stabla

# Samopodešavajuća stabla

- Binarna stabla – brz pristup podacima, ali komplicirano održavanje – dodavanje i brisanje je sporo
- **Samopodešavajuća stabla** – podatke kojima se češće pristupa podići na više razine (prisjetite se samopodešavajućih listi s prethodnog predavanja)
- → Svaki čvor mora „znati” koliko mu se puta pristupilo
- Dvije osnovne strategije:
  - 1) *Single rotation* – čvor kojem se pristupilo rotirati oko roditelja
  - 2) *Moving to the Root* – čvor kojem se pristupilo postaje korijen stabla



# Za kraj...

## Korisni linkovi:

- Vizualizacija mnogih algoritama koje ćemo učiti na ovom predmetu:
  - <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>
  - → Primjerice, dodavanje i brisanje za AVL stablo:
    - <https://www.cs.usfca.edu/~galles/visualization/AVLtree.html>
    - → **korisno za vježbanje i provjeru rješenja!**
- Online alat za izradu grafova:
  - <https://www.draw.io/>
  - → Intuitivan i jednostavan za korištenje