



# ALGORITMI I STRUKTURE PODATAKA

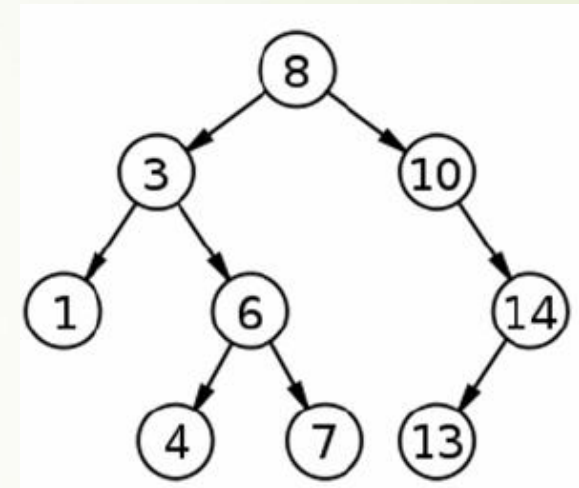
RAČUNSKE VEŽBE – TERMIN BR. 8 – BINARNA STABLA – prvi deo

ALDINA AVDIĆ, DIPL. INŽ. – [apl.jaskovic@np.ac.rs](mailto:apl.jaskovic@np.ac.rs)

RAČUNARSKA TEHNIKA, SOFTVERSKO INŽENJERSTVO, INFORMATIKA I MATEMATIKA

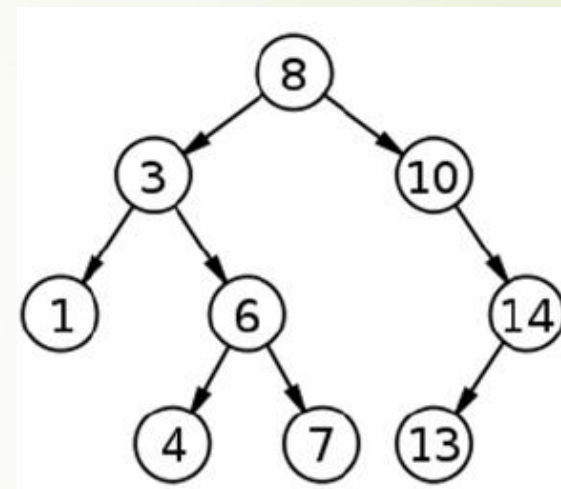
# Binarno stablo

- × **Čvor stabla** je jedna memorijska ćelija stabla. Kod binarnog stabla on može imati nula, jedan ili dva podčvora.
- × **Koren stabla** je čvor stabla koji nije potomak nijednog drugog čvora u stablu.
- × **List** je čvor stabla koji nema ni jedan podčvor.
- × **Roditelj** nekog čvora je čvor koji pokazuje na njega.
- × **Dete** nekog čvora je čvor na koji neki drugi čvor pokazuje.
- × **Podstablo** ili **podgrana** je skup svih čvorova stabla koji se nalaze levo ili desno od nekog od čvorova stabla.



# Binarno stablo

- × U vezi slike: **čvorovi** stabla su elementi prikazani krugovima.
- × Upisani brojevi su vrednosti ključeva po kojima se elementi sortiraju.
- × **Čvor** sa ključem 8 je koren stabla.
- × Njegova deca su **čvorovi** sa ključevima 3 i 10.
- × Roditelj **čvorova** sa vrednošću ključeva 3 i 10 je **čvor** sa ključem 8.
- × Listovi stabla su **čvorovi** sa ključevima 1, 4, 7 i 13.



# Binarno stablo

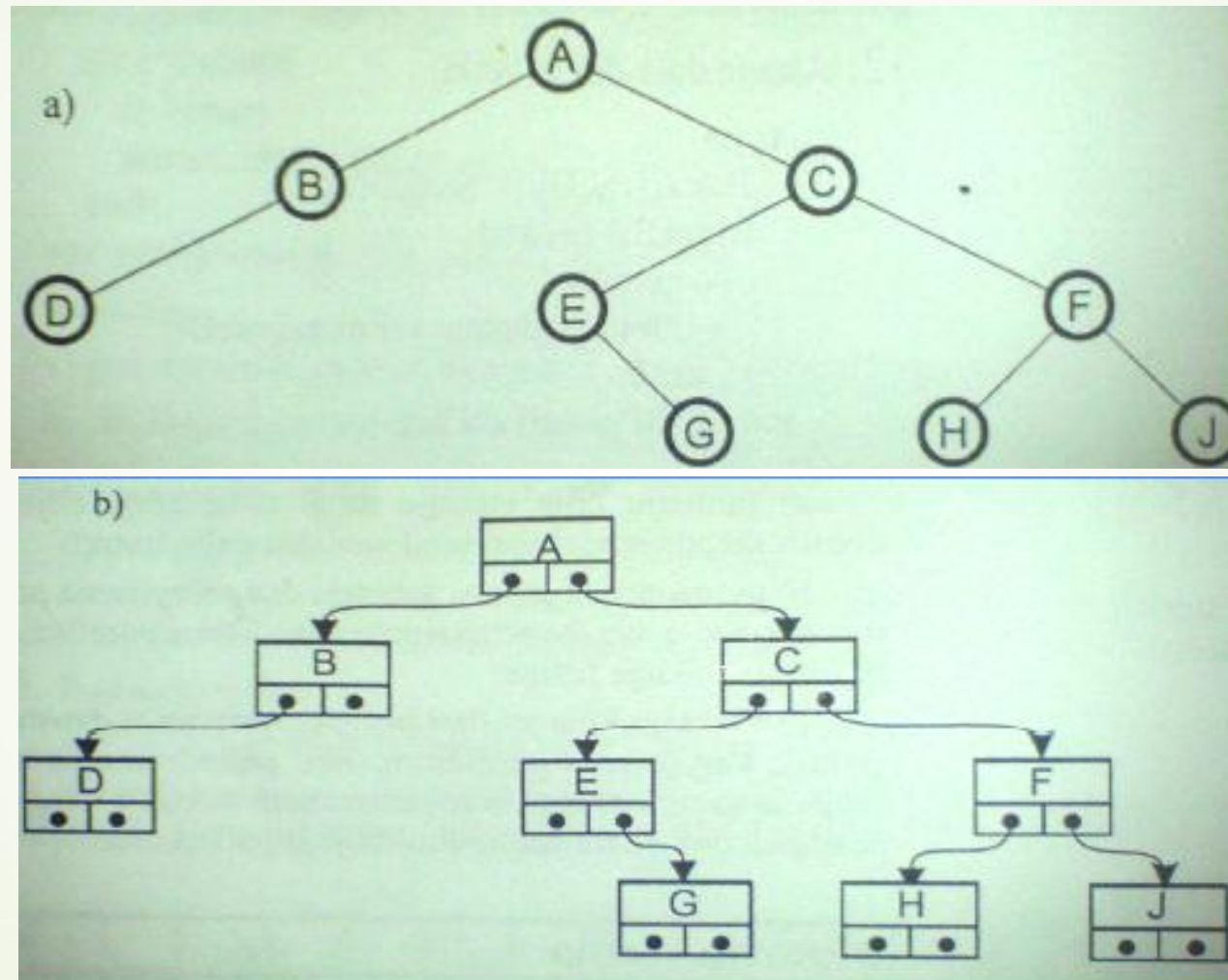
- × Binarno stablo je struktura koja omogućava efikasno lociranje, brisanje i umetanje slogova.
- × Definiše se kao konačan skup elemenata koji može biti prazan ili sadržati dva podstabla koja nazivamo levo i desno podstablo.
- × Definiše se rekurzivno.
- × Deklaracija binarnog stabla:

```
#include<stdlib.h>
#include<stdio.h>

struct bin_tree {
    int data;
    struct bin_tree * right, * left;
};

typedef struct bin_tree node;
```

# Binarno stablo - reprezentacija



# Obilazak stabla

- × Za rad sa binarnim stablima postoji mnoštvo algoritama, ali svi oni se oslanjaju na osnovnu ideju - obilazak čvorova stabla.
- × Za obilazak čvorova (ako stablo nije prazno) mogu se koristiti sledeća tri načina koja se definišu rekurzivno:
  - × INORDER: LEVO PODSTABLO, KOREN, DESNO PODSTABLO (LKD)
  - × POSTORDER: LEVO PODSTABLO, DESNO PODSTABLO, KOREN (LDK)
  - × PREORDER: KOREN, LEVO PODSTABLO, DESNO PODSTABLO (KLD)

# Obilazak stabla – zadaci

× **Zadatak 1.** Ispisati elemente binarnih stabla prikazanih na slici u redosledima LKD, LDK i KLD.

× (a)

× LKD: D B A E G C H F I

× LDK: D B G E H I F C A

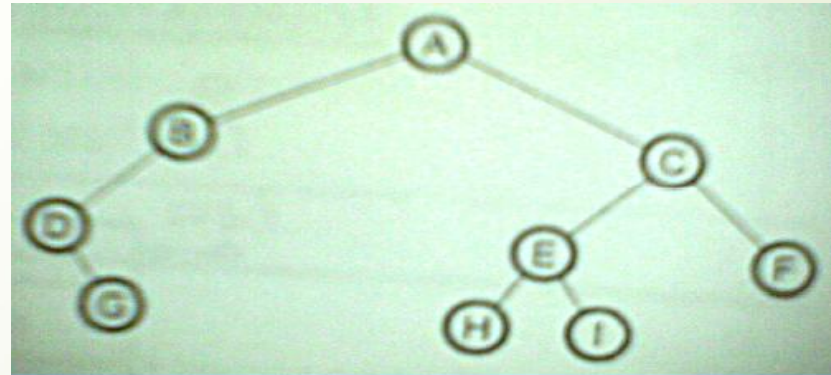
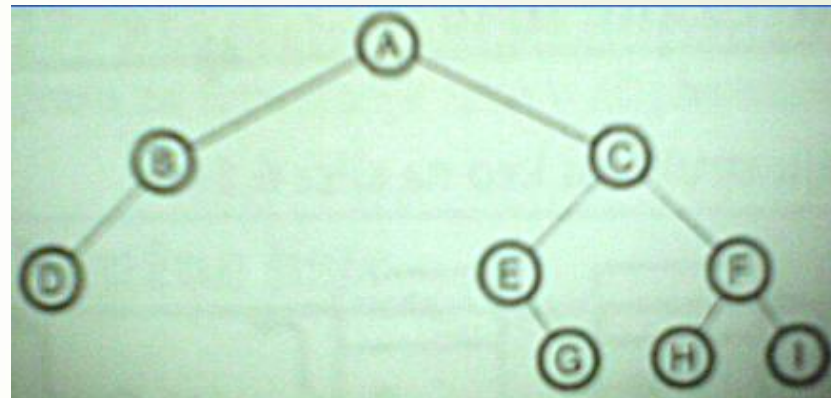
× KLD: A B D C E G F H I

× (b)

× LKD: D G B A H E I C F

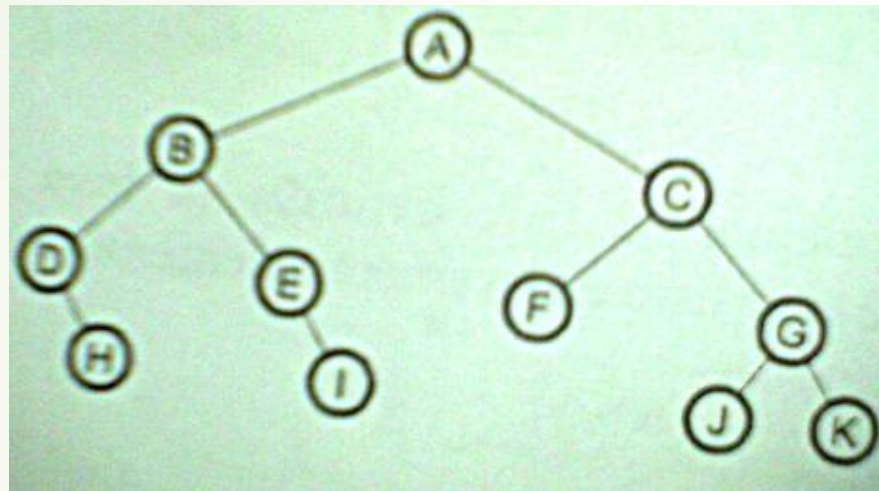
× LDK: G D B H I E F C A

× KLD: A B D G C E H I F

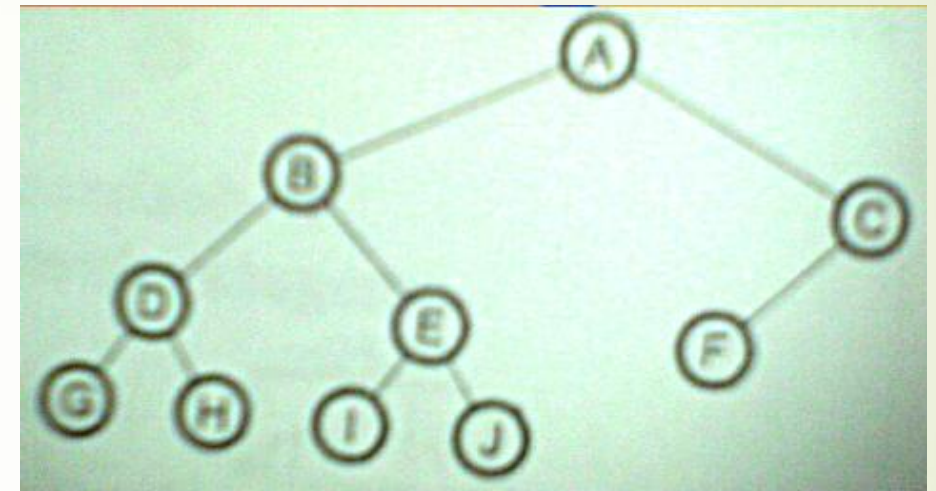




# Za samostalni rad



(c)



(d)



# Obilazak stabla - zadatak 2

- × Šta je **preorder**, a šta **inorder** obilazak stabla? Ako za jedno binarno stablo **preorder** obilazak daje poredak ATNEIFCSBDGPMLK, a **inorder** obilazak daje poredak EINSFCBTGPDLMKA, rekonstruisati izgled ovog stabla i objasniti postupak.
- × Obilazak stabla: sistematično posećivanje svih čvorova samo jednom
- × Postoji više načina da se obiđe stablo:
  - × preorder
  - × inorder
  - × postorder

# Obilazak stabla - zadatak 2

## Preorder

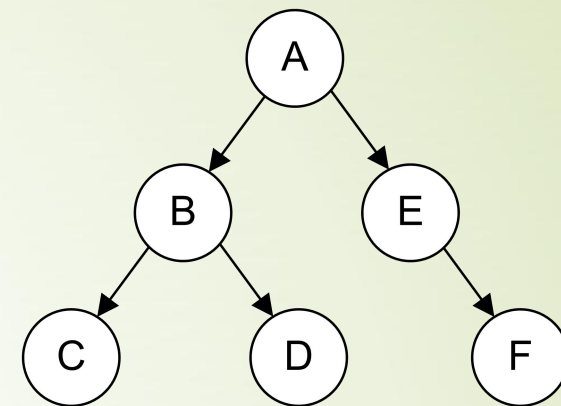
1. Poseti se koren
2. Obiđe se levo podstablo na *preorder* način
3. Obiđe se desno podstablo na *preorder* način

## Inorder

1. Obiđe se levo podstablo na *inorder* način
2. Poseti se koren
3. Obiđe se desno podstablo na *inorder* način

## Postorder

1. Obiđe se levo podstablo na *postorder* način
2. Obiđe se desno podstablo na *postorder* način
3. Poseti se koren



ABCDEF

← Prvi čvor je koren

CBDAEF

← Prvi čvor je "najlevlji"  
čvor u stablu

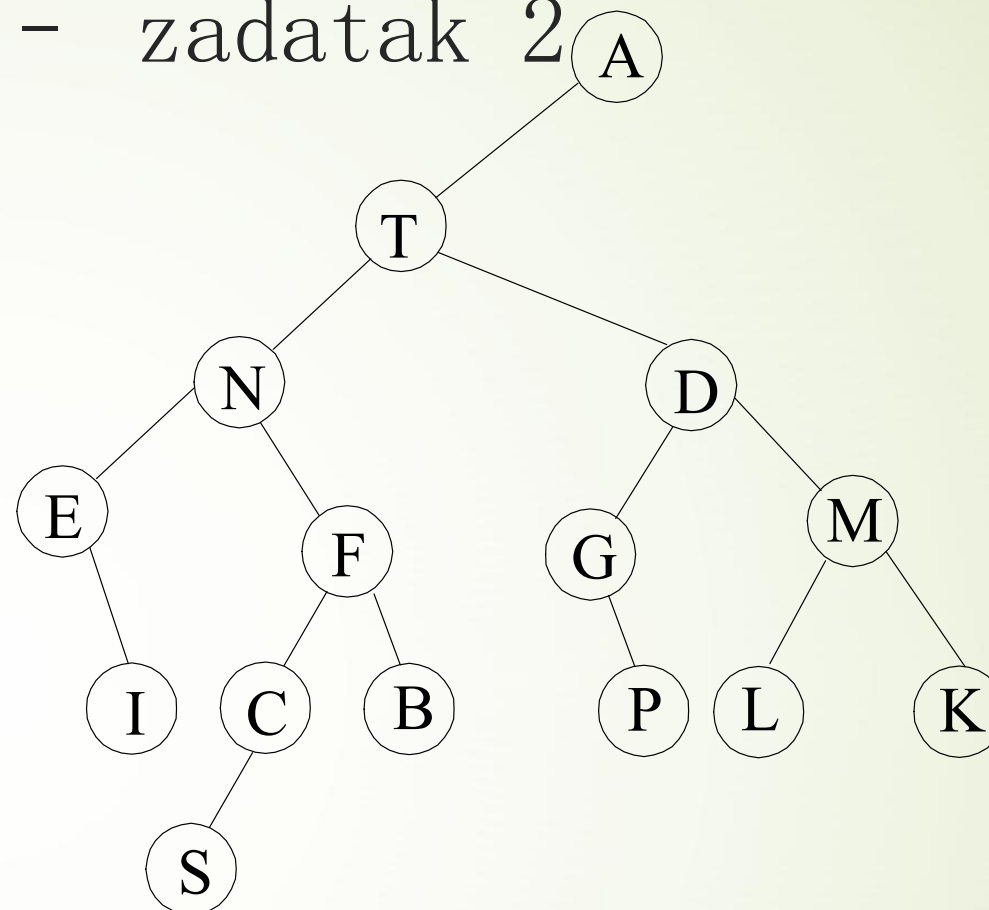
CDBFEA

← Poslednji čvor je koren

# Obilazak stabla - zadatak 2

11

preorder: ATNEIFCSBDGPMLK  
inorder: EINSFCBTGPDLMKA



## Početak rekonstrukcije stabla:

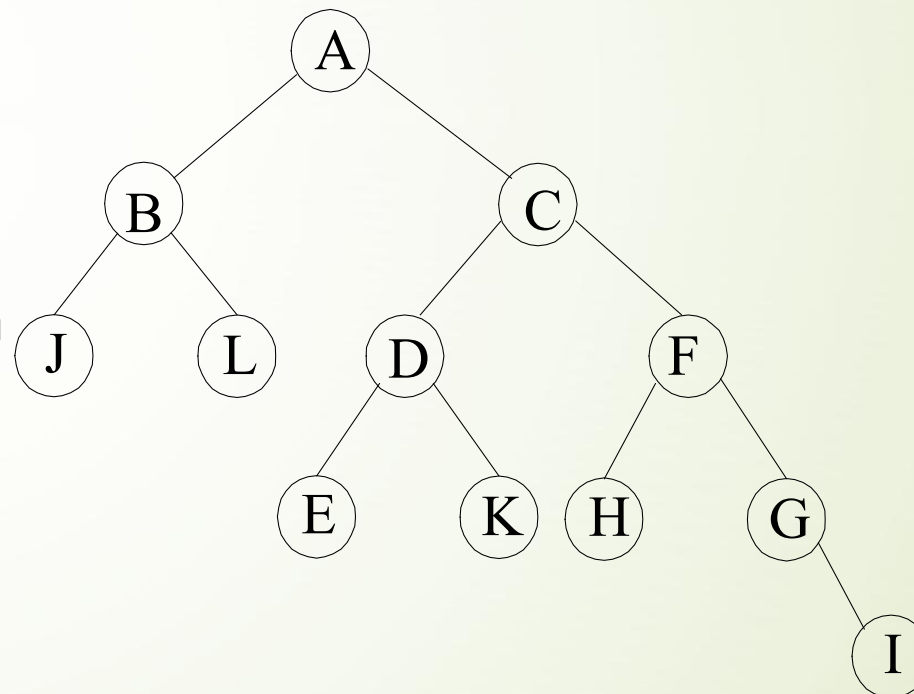
- prvi čvor u *inorder* poretku, E, “najlevlji” je čvor stabla. Prati se *preorder* poredak sve dok se ne dođe do čvora E: A-T-N-E. Sigurno je da je T levi potomak čvora A, N, levi potomak čvora T i E levi potomak čvora N.
- u *inorder* poretku, pre čvora N dolazi čvor I – iz toga se zaključuje da je I desni potomak čvora E.

## Obilazak stabla - zadatak 3

Šta je *inorder*, a šta *postorder* obilazak stabla? Ako za jedno binarno stablo *inorder* obilazak daje poredak **JBLAEDKCHFGI**, a *postorder* obilazak daje poredak **JLBEKDHIGFCA**, rekonstruisati izgled ovog stabla i objasniti postupak.

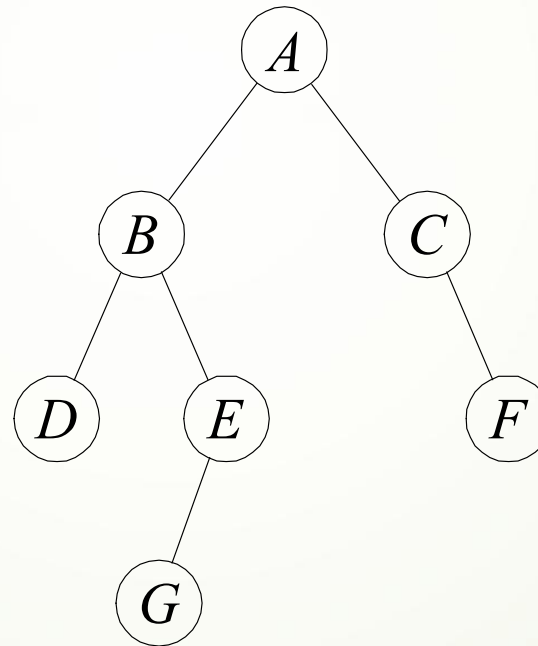
### Postupak:

- u *postorder* poretku, poslednji čvor je A, koji je sigurno koren stabla
- iz *inorder* poretka zaključujemo da su čvorovi J, B i L u levom podstablu korena, a ostali u desnom
- dakle, C je koren desnog podstabla (preposlednji čvor u *postorder* poretku), a B je koren levog podstabla („najdešnji“ čvor od čvorova J, B i L u *postorder*)



# Obilazak stabla - zadatak 4

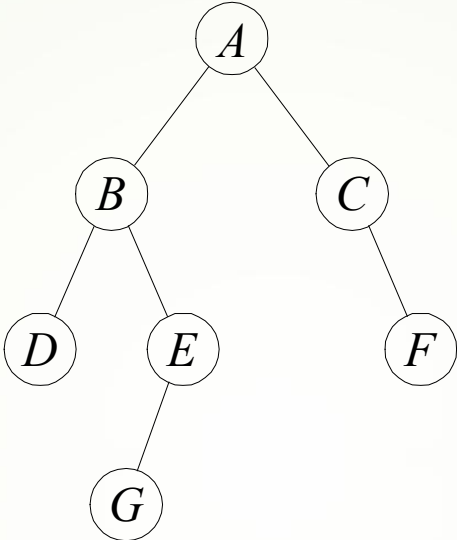
Skicirati i objasniti iterativnu realizaciju inorder algoritma obilaska binarnog stabla. Ilustrovati rad algoritma po koracima na primeru sledećeg stabla.



# Obilazak stabla - zadatak 4

14

```
INORDER-I(root)
  next = root
  loop
    while (next ≠ nil) do
      PUSH(S, next)
      next = left(next)
    end_while
    if (not STACK-EMPTY(S)) then
      next = POP(S)
      P(next)
      next = right(next)
    else
      return
    end_if
  end_loop
```



<i>next</i>	<i>stack</i>	<i>inorder</i> poredak
<i>A</i>	<i>A</i>	-
<i>B</i>	<i>AB</i>	-
<i>D</i>	<i>ABD</i>	-
nil	<i>AB</i>	<i>D</i>
nil	<i>A</i>	<i>DB</i>
<i>E</i>	<i>AE</i>	<i>DB</i>
<i>G</i>	<i>AEG</i>	<i>DB</i>
nil	<i>AE</i>	<i>DBG</i>
nil	<i>A</i>	<i>DBGE</i>
nil		<i>DBGEA</i>
<i>C</i>	<i>C</i>	<i>DBGEA</i>
nil		<i>DBGEAC</i>
<i>F</i>	<i>F</i>	<i>DBGEAC</i>
nil		<i>DBGEACF</i>
nil		



## Huffman-ov algoritam - Zadatak 5

U poruci se javlja 8 simbola (A, B, C, D, E, F, G, H) sa datim verovatnoćama pojavljivanja.

Prikazati postupak izbora optimalnih prefiksnih kodova primenom standardnog **Huffman**-ovog algoritma.

Simboli	A	B	C	D	E	F	G	H
Verovatnoće	8	6	13	12	23	10	18	10

Napomena: verovatnoće su izražene u procentima

## Zadatak 5 – rešenje

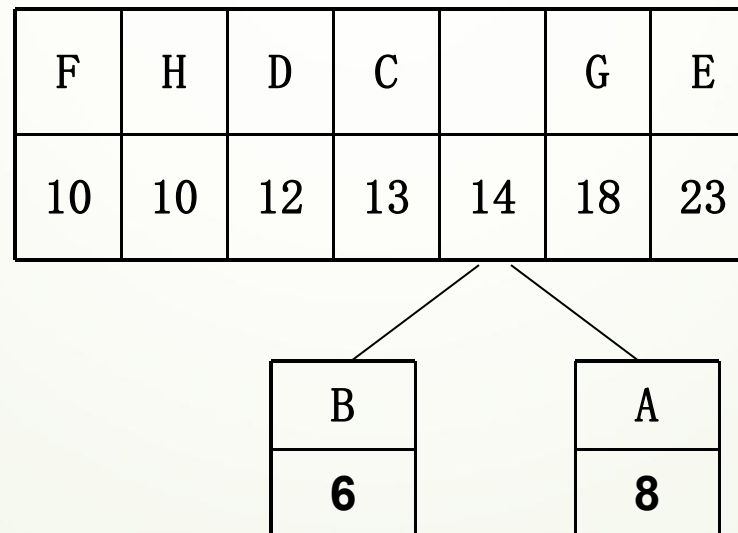
- × Simboli se najpre urede po rastućoj vrednosti verovatnoće pojavljivanja.

Simboli	B	A	F	H	D	C	G	E
Verovatnoće	6	8	10	10	12	13	18	23

- Svaki od prikazanih simbola treba shvatiti kao nezavisno stablo sa jednim čvorom (tj. korenom).
- Time je dobijena šuma

# Zadatak 5 – rešenje

- × Huffman-ov algoritam:
  - × od dva stabla čiji koreni imaju najmanju težinu (verovatnoću), formirati jedno stablo
  - × težina (verovatnoća) korena tog stabla je zbir težina (verovatnoća) korenova stabala od kojih je to stablo nastalo
  - × novo stablo se vraća u skup stabala koji se ponovo uredi
  - × postupak se ponavlja dok se ne formira jedinstveno stablo



# Zadatak 5 – rešenje

<b>D</b>	<b>C</b>		<b>G</b>		<b>E</b>
<b>12</b>	<b>13</b>	<b>14</b>	<b>18</b>	<b>20</b>	<b>23</b>

<b>B</b>	<b>A</b>	<b>F</b>	<b>H</b>
<b>6</b>	<b>8</b>	<b>10</b>	<b>10</b>

	<b>G</b>		<b>E</b>	
<b>14</b>	<b>18</b>	<b>20</b>	<b>23</b>	<b>25</b>

<b>B</b>	<b>A</b>	<b>F</b>	<b>H</b>	<b>D</b>	<b>C</b>
<b>6</b>	<b>8</b>	<b>10</b>	<b>10</b>	<b>12</b>	<b>13</b>

	<b>E</b>		
<b>20</b>	<b>23</b>	<b>25</b>	<b>32</b>

<b>F</b>	<b>H</b>
<b>10</b>	<b>10</b>

<b>D</b>	<b>C</b>		<b>G</b>
<b>12</b>	<b>13</b>	<b>14</b>	<b>18</b>

<b>B</b>	<b>A</b>
<b>6</b>	<b>8</b>

<b>25</b>	<b>32</b>	<b>43</b>

<b>D</b>	<b>C</b>
<b>12</b>	<b>13</b>

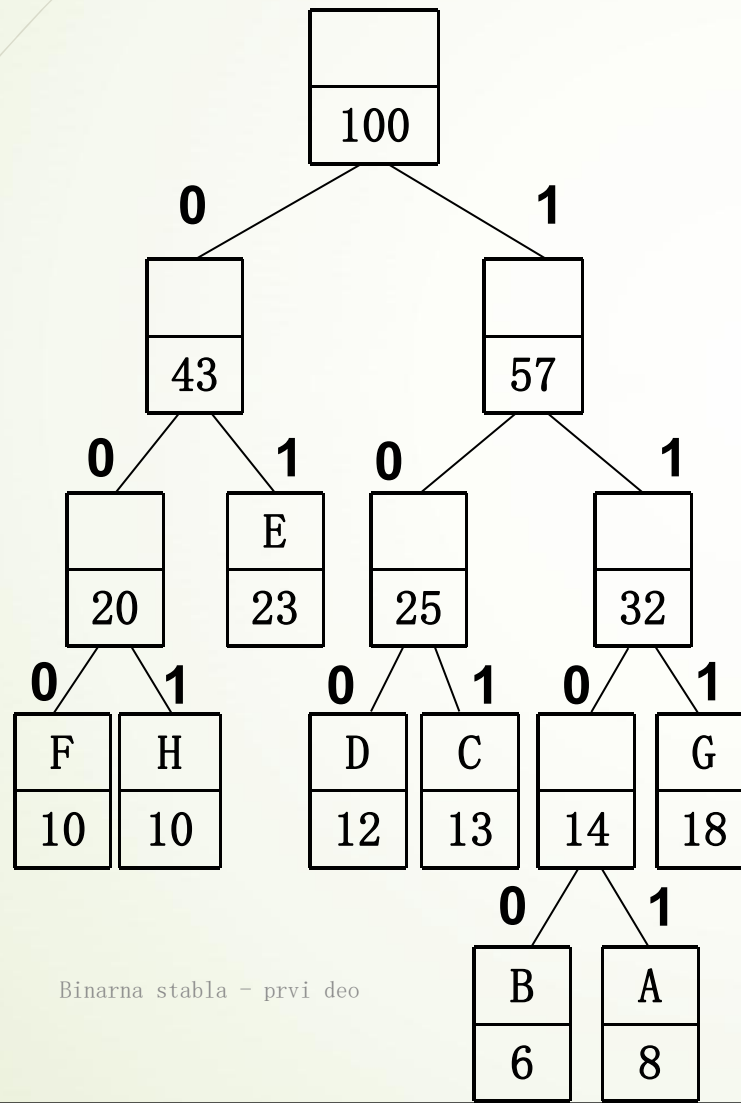
	<b>G</b>
<b>14</b>	<b>18</b>

	<b>E</b>
<b>20</b>	<b>23</b>

<b>B</b>	<b>A</b>
<b>6</b>	<b>8</b>

<b>F</b>	<b>H</b>
<b>10</b>	<b>10</b>

# Zadatak 5 – rešenje



Binarna stabla – prvi deo

- ✗ Nakon konstruisanja stabla, grane treba obeležiti
- ✗ Na primer, sve leve grane dobijaju identifikator 0, sve desne identifikator 1
- **Kôd nekog simbola predstavlja putanju od korena do lista u kome se nalazi dati simbol.**

Simboli	A	B	C	D	E	F	G	H
Kodovi	1101	1100	101	100	01	000	111	001

## Zadatak 6

U jednom prenosnom sistemu poruke se sastoje od simbola A, B, C, D, E, F, G i H sa verovatnoćama pojavljivanja 0.29, 0.25, 0.2, 0.12, 0.05, 0.04, 0.03 i 0.02, respektivno. Kodirati simbole tako da prosečna dužina prenesene poruke bude minimalna i izračunati ovu dužinu. Obrazložiti postupak.

Simboli	A	B	C	D	E	F	G	H
Verovatnoće	0.29	0.25	0.2	0.12	0.05	0.04	0.03	0.02



## Zadatak 6 – Rešenje

Simboli	A	B	C	D	E	F	G	H
Verovatnoće	0.29	0.25	0.2	0.12	0.05	0.04	0.03	0.02

Simboli	A	B	C	D	E	F	G	H
Kodovi	11	01	00	100	10111	10110	10101	10100

Prosečna dužina:

$$2 \cdot 0.29 + 2 \cdot 0.25 + 2 \cdot 0.2 + 3 \cdot 0.12 + 5 \cdot 0.05 + 5 \cdot 0.04 + 5 \cdot 0.03 + 5 \cdot 0.02 = 2.54 \text{ bita}$$

Kada bi se kodiranje vršilo bez uvida u verovatnoće, prosečna dužina bi bila 3 bita

## Zadatak 7 – Za samostalni rad

U jednom prenosnom sistemu poruke se sastoje od simbola A, B, C, D, E, F, G i H sa verovatnoćama 0.18, 0.36, 0.15, 0.17, 0.07, 0.02, 0.03 i 0.02, respektivno. Kodirati simbole tako da prosečna dužina prenesene poruke bude minimalna i izračunati ovu dužinu. Obrazložiti postupak.

Simboli	A	B	C	D	E	F	G	H
Verovatnoće	0.18	0.36	0.15	0.17	0.07	0.02	0.03	0.02

Rešenje: 2.53 bita

# Test

1. Šta je koren stabla?
2. Šta je podstablo stabla?
3. Kako se može implementirati stablo?
4. Šta je list stabla?
5. Koje vrste obilazaka stabla postoje?
6. U čemu se razlikuju obilasci stabla?
7. Napisati sve vrste obiliskaza stabla za stablo na slajdu 2.
8. Objasniti delove stabla sa slajda 5.
9. Napisati korake koji nedostaju u rešavanju zadatka 6.
10. Napisati korake koji nedostaju u rešavanju zadatka 7.

# Test

- × Test poslati do 27.04.2020. u 14h na mejl [apl.jaskovic@np.ac.rs](mailto:apl.jaskovic@np.ac.rs) prema uputstvima sa sajta univerziteta



Hvala na pažnji!