


Algoritmi i programiranje

Tipovi i strukture
podataka



Tipovi i strukture podataka

- U matematici je uobičajeno da se promenljive klasifikuju na osnovu neke ključne osobine
 - prave se jasne razlike izmedju realnih, kompleksnih ili logičkih promenljivih
 - izmedju promenljivih koje predstavljaju individualne (skalarne) vrednosti, ili skup vrednosti, ili skup skupova,...
- Ovakav način klasifikacije promenljivih je jednako važan kada je u pitanju rešavanje problema uz pomoć računara!

Šta je definisano tipom podatka?

- Svaka konstanta, promenljiva ili izraz koji se pojavljuju u programu pripadaju određenom tipu (klasi).
- Tipom podatka je određeno:
 1. Skup vrednosti koje konstante, promenljive ili izrazi mogu uzimati;
 2. Skup operatora (operacija) koje se mogu primenjivati nad konstantama i promenljivima određenog tipa;
 3. način predstavljanja u memoriji računara

Osnovni tipovi podataka

- Brojni (numerički)
- Logički
- Tekstualni (znakovni)

Numerički tipovi:

- Celobrojni (integer)
- Realni
- (kompleksni)

Celobrojni tip (integer)

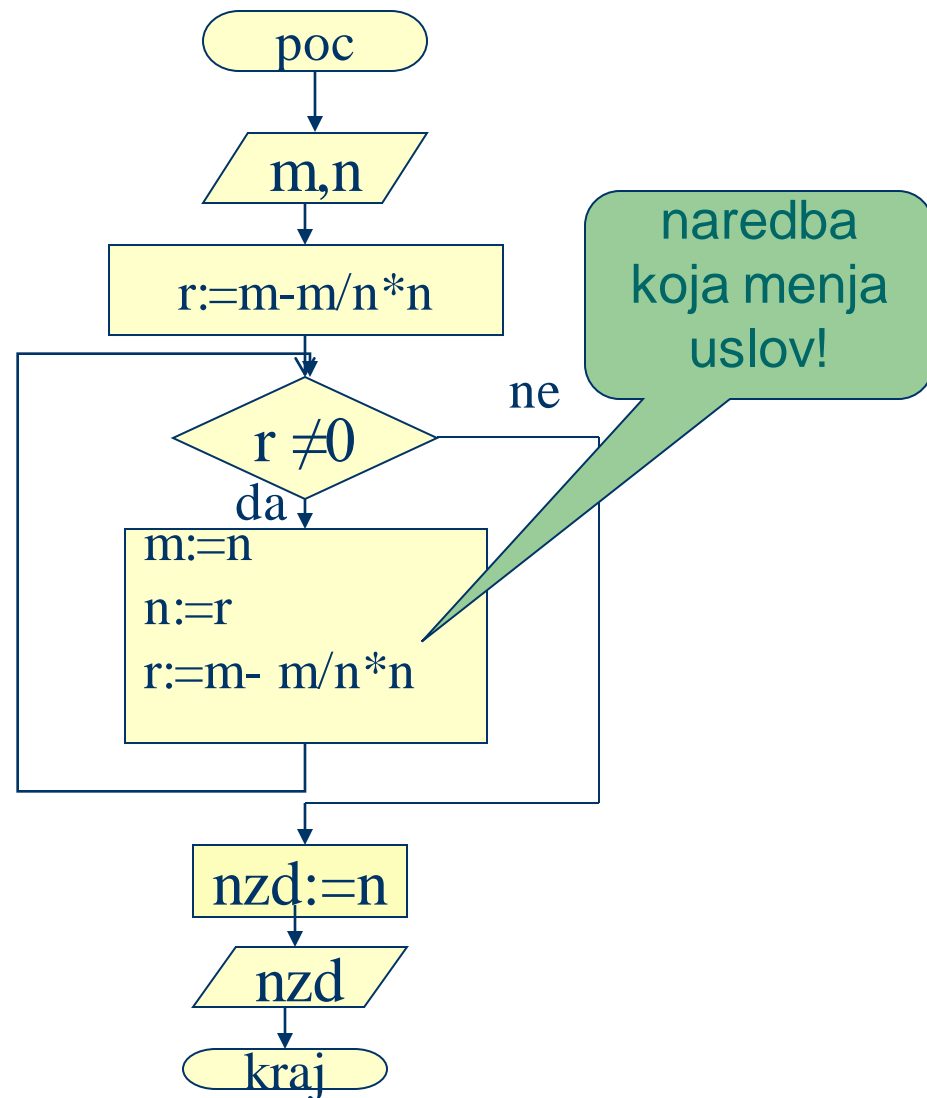
- Podaci celobrojnog tipa mogu uzimati celobrojne pozitivne i negativne brojne vrednosti;
 - opseg vrednosti zavisi od broja bitova koji se koristi za predstavljanje podataka
 - Ako se koristi m bitova, onda je opseg vrednosti -2^{m-1} do $2^{m-1}-1$
 - brojevi van ovog opsega ne pripadaju tipu **integer**

Celobrojni tip (integer)

- Najčešća vrednost za $m=16$,
 - opseg -32768 do 32767
- Operacije koje se mogu izvršavati nad podacima celobrojnog tipa su +, -, *, /
- kada se nad dva podatka celobrojnog tipa primeni neki od operatora, dobija se rezultat celobrojnog tipa!

Celobrojni tip

- $r := \text{"ostatak od } m/n\text{"}$?
- $r := m - m/n * n$



Realni tip

- Koristi se za predstavljanje realnih brojnih vrednosti.
- Podaci realnog tipa se predstavljaju u normalizovanom obliku:

$$m * b^e, \quad \frac{1}{b} \leq m < 1$$

- celi deo je jednak 0, a prva cifra iza decimalne tačke je značajna cifra brojnog sistema

- 0.1254E+3

125.4

0.01254E+4

nije normalizovani
oblik

Realni tip

- Opseg vrednosti zavisi od broja cifara za predstavljanje eksponenta
- Preciznost zavisi od broja cifara za predstavljanje mantise broja
- Operacije $+$, $-$, $*$, $/$

Logički tip

- Podaci ovog tipa mogu uzimati samo dve vrednosti – “**istina**” (tačno) i “**laž**” (netačno)
- “istina” se najčešće predstavlja svim jedinicama u registru računara;
- “laž” se predstavlja svim nulama u registru;
- Operacije: **I** (konjunkcija), **ILI** (disjunkcija),
 NE (negacija)

Znakovni tip

- Koristi se za predstavljanje tekstualnih podataka:
 - podaci ovog tipa se sastoje od velikih i malih slova abecede, decimalnih cifara, specijalnih znakova, znakova interpunkcije i blanko znaka
 - Svaki znak je kodiran odgovarajućim kodom (ASCII, EBCDIC,...)
 - Podskupovi slova i cifara su uredjeni i važi da je

“A” < “B” < “C” < ... < “Z”

“0” < “1” < ... < “9”

Znakovni tip

- Nad podacima znakovnog tipa moguće je vršiti operaciju poredjenja i operaciju nadovezivanja (konkatenacije)

“AAB” < “AB” (poredjenje)

“AB” # “C”  “ABC” (nadovezivanje)

Strukturni tipovi

- Dobijaju se od osnovnih tipova (ili prethodno definisanih struktura) definisanjem veza izmedju elemenata strukture.
 - **Strukture podataka mogu biti:**
 1. **Linearne** – svaki element strukture, izuzev prvog i poslednjeg, je u relaciji samo sa dva druga elementa, prethodnikom i sledbenikom.
 - polja
 - linearne lančane liste
 - magacini
 - redovi
 2. **Nelinearne** – veze izmedju elemenata strukture su proizvoljne (svaki element može imati veći broj prethodnika ili sledbenika).
 - stabla
 - grafovi

Polja (nizovi)

- Polje je homogena struktura podataka (sastavljena od elemenata istog tipa) koji imaju zajedničko ime. Mogu biti:
 - jednodimenzionalna (vektori),
 - dvodimenzionalna (matrice) i
 - višedimenzionalna polja.
- Elementu polja se pristupa navodjenjem zajedničkog imena i uredjene n-torke indeksa:

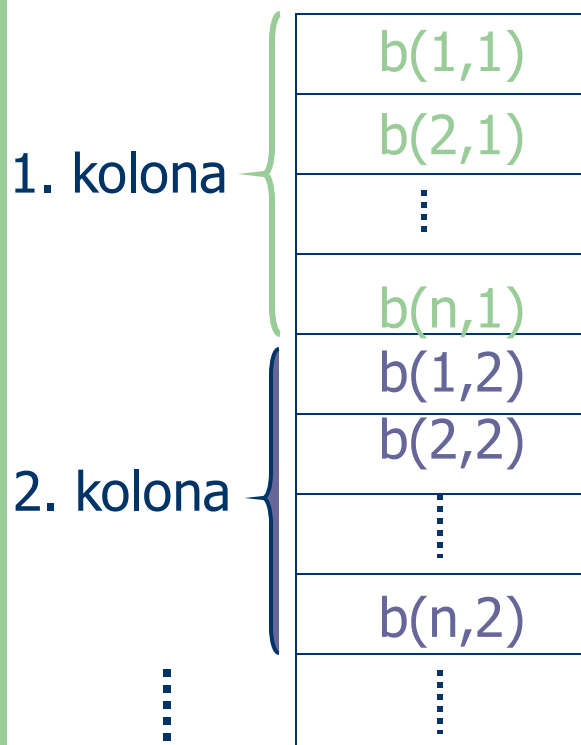
$A(i_1, i_2, \dots, i_n)$

- Elementi polja se smestaju u sukcesivne (susedne) memorijske lokacije

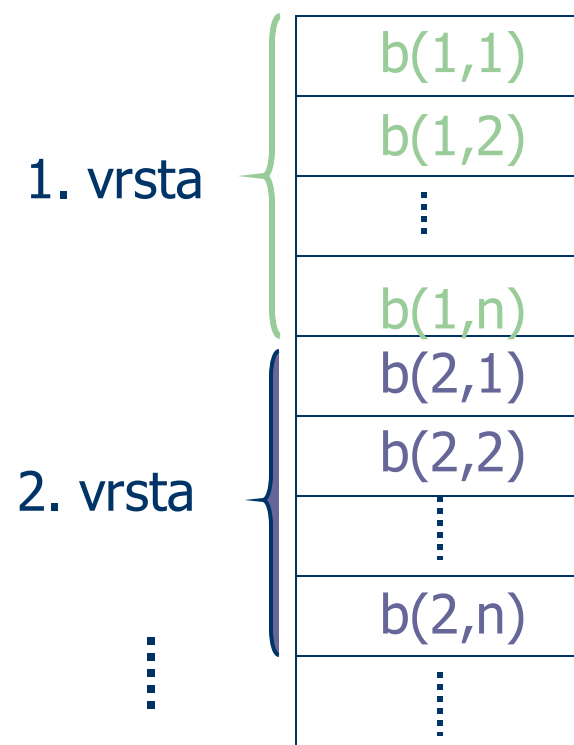
i	a(1)
i+1	a(2)
i+2	a(3)
i+3	a(4)
...	⋮

Polja (nast.)

- Kod smeštanja dvodimenzionalnih polja u memoriju vrši se linearizacija (po vrstama ili kolonama).



ili

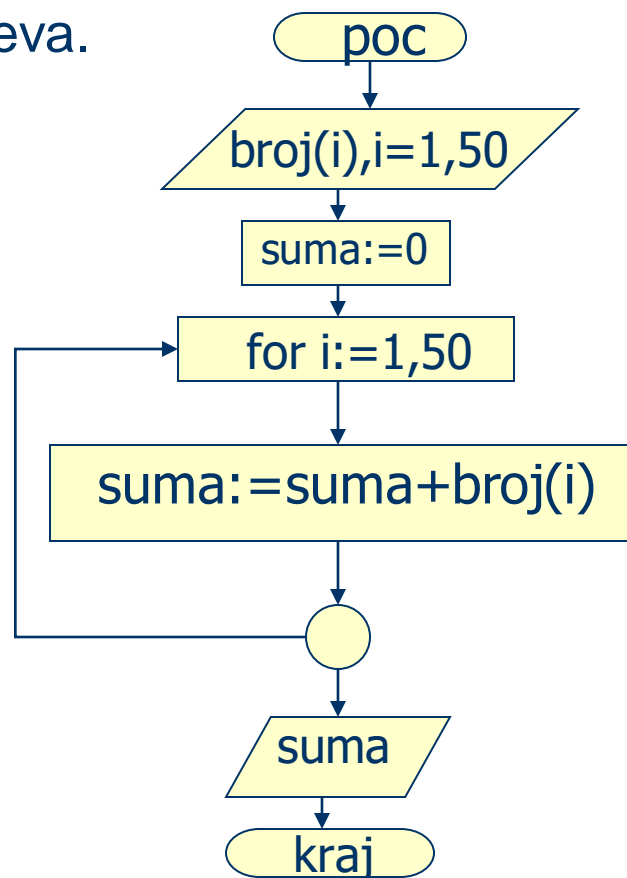


Polja (nast.)

- Polja su statičke strukture podataka:
 - dimenzije polja se ne mogu menjati u toku izvršenja programa,
 - ubacivanje ili izbacivanje elementa iz polja zahteva potpuno preuredjenje polja (na vežbama!).
- Direktan pristup elementima (navodjenjem indeksa).

Polja – Primer 1

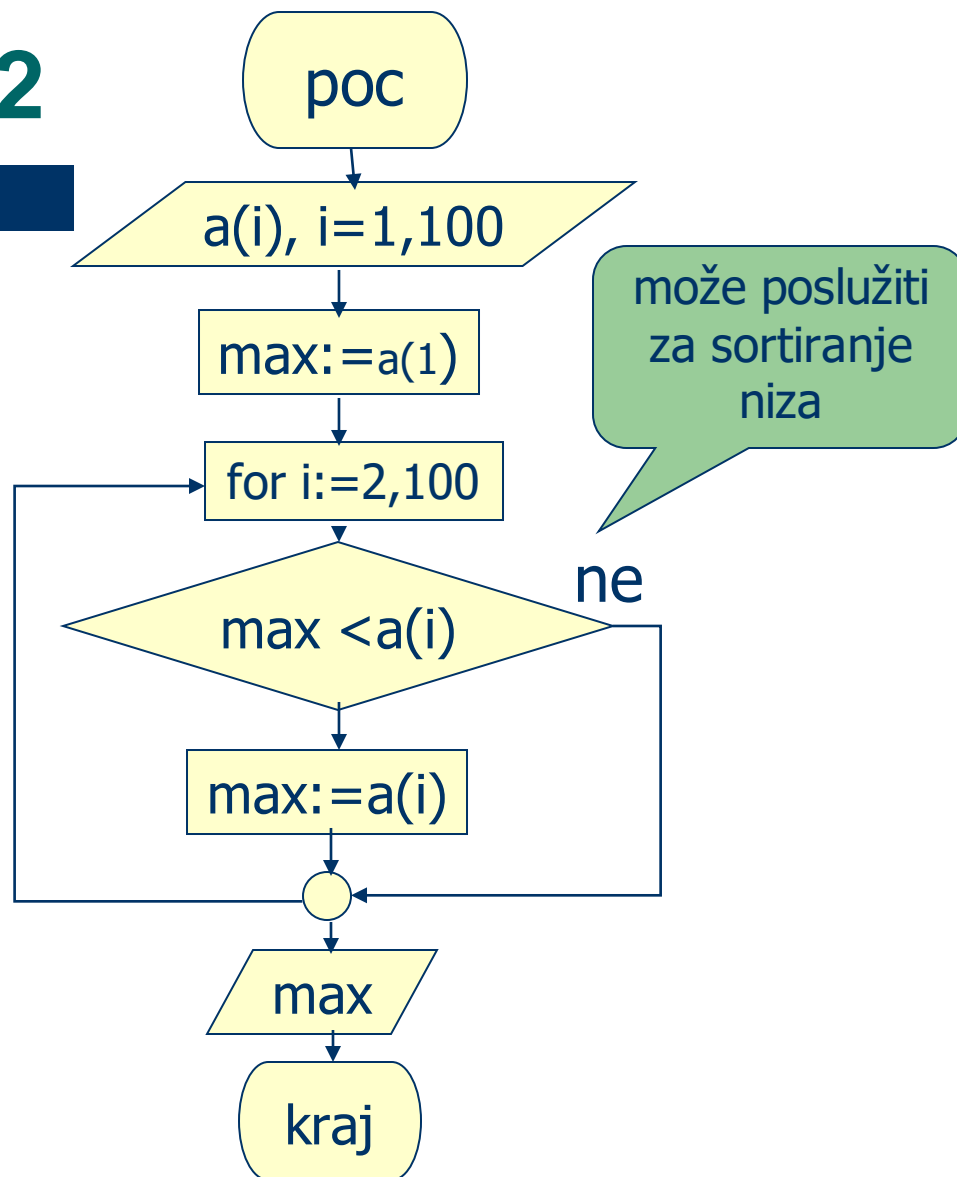
- Zadata je lista od 50 brojeva.
Naći sumu svih brojeva.



kako naći
proizvod?

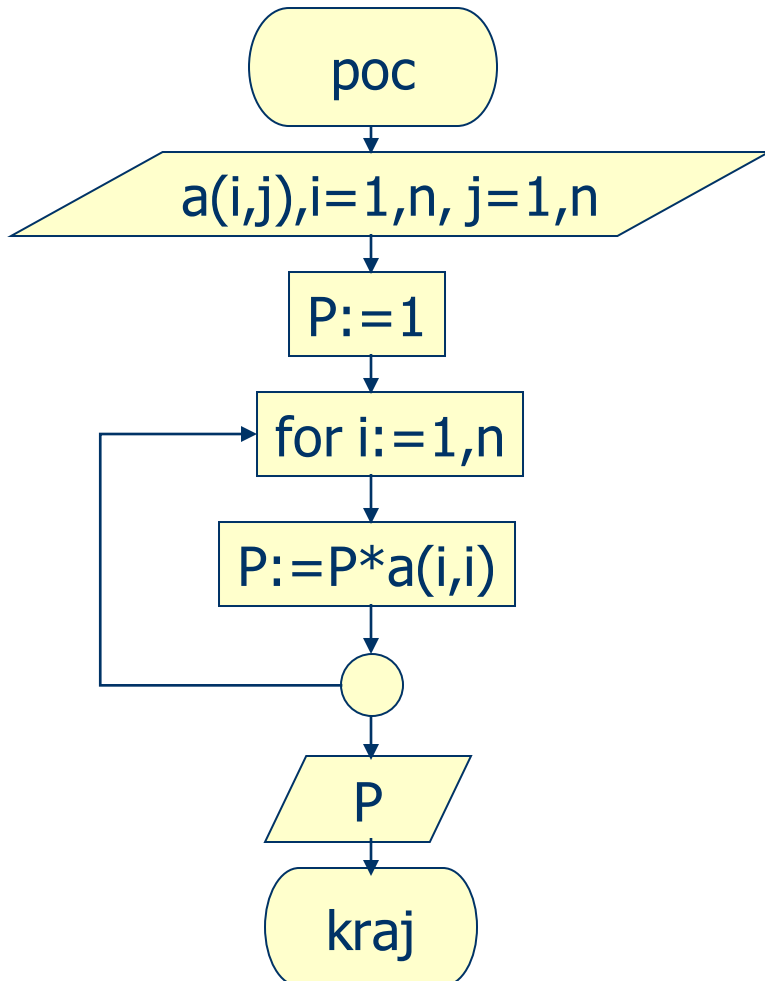
Polja – Primer 2

- Naći maksimalni element u nizu od 100 brojeva.

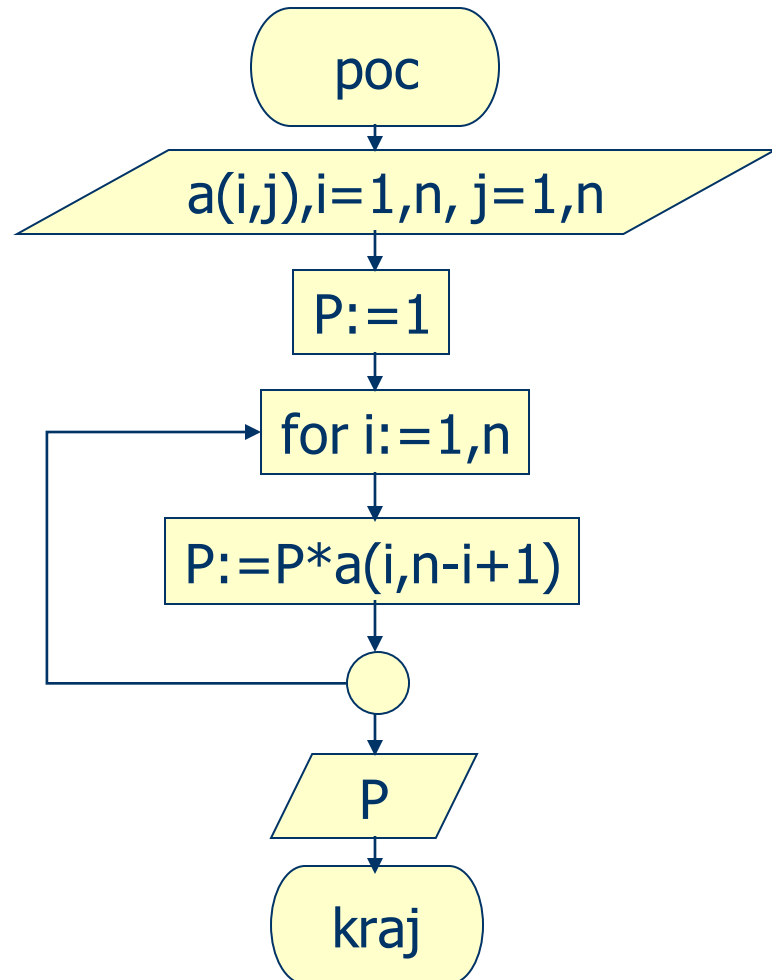


Polja – Primer 3

- Naći proizvod elemenata na glavnoj dijagonali matrice $A_{n \times n}$



- Naći proizvod elemenata na sporednoj dijagonali matrice $A_{n \times n}$



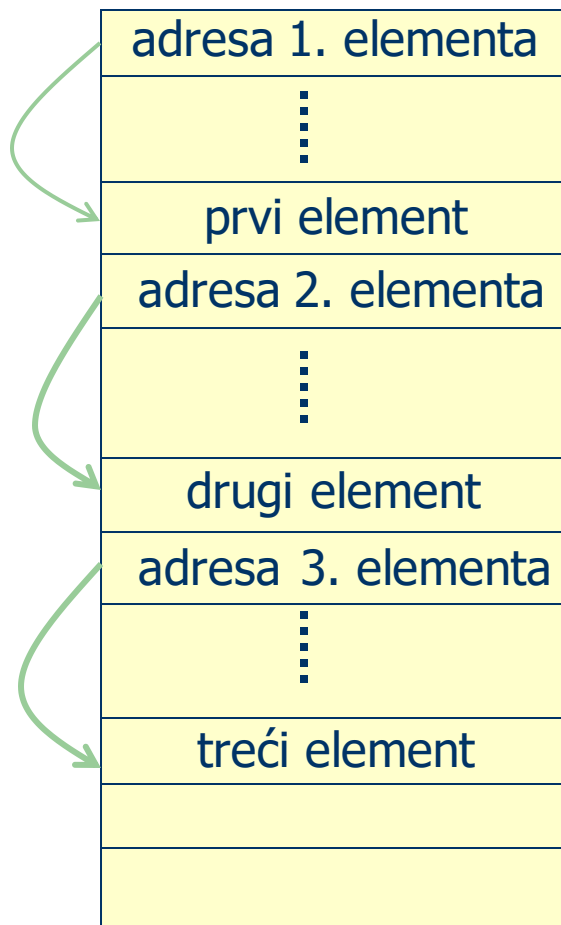
Linearne lančane liste

- Svaki element strukture se sastoji iz dva dela:
 - Informacionog (sadrži podatak) i
 - Pokazivačkog (sadrži adresu sledećeg elementa liste).



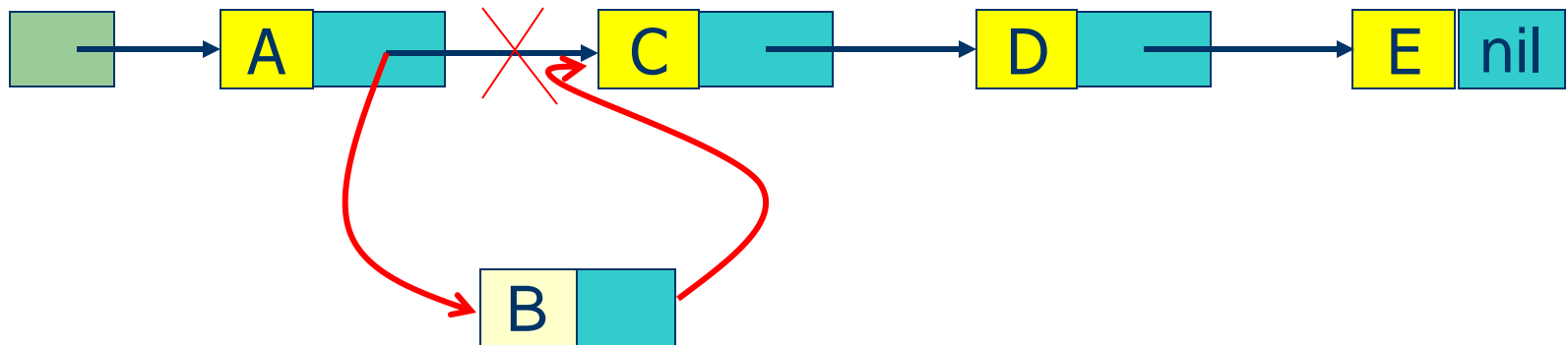
- Ne zahteva sukcesivne memorijske lokacije za pamćenje elemenata structure.
- Dinamička struktura – dimenzije liste se mogu menjati u toku izvršenja programa.

Linearne lančane liste

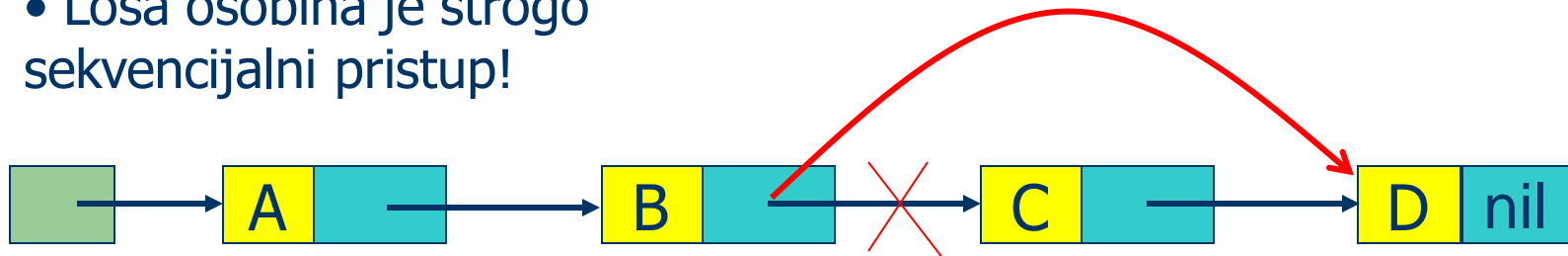


Linearne lančane liste

- Ubacivanje i izbacivanje elemenata ne zahteva kompletno preuredjenje strukture.



- Loša osobina je strogo sekvencijalni pristup!



Magacin (LIFO – Last-In-First-Out)

- Linearna struktura kod koje se upis i čitanje podataka obavlja na jednom kraju koji se zove vrh magacina:
 - po ovom principu se slažu korpe u samoposluži, poslužavnici u ekspres restoranu,...



inicijalno je $\text{vrh} := 0$

```
if (vrh < n) then  
  vrh := vrh + 1  
  mag(vrh) := podatak  
endif
```

realizacija
magacina pomoću
polja (vektora)

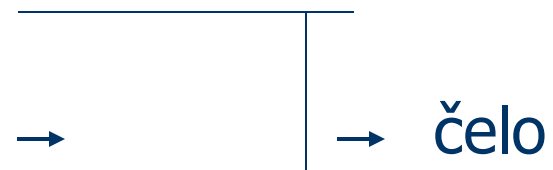
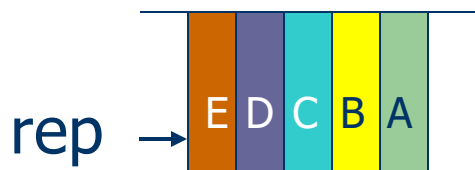
```
if (vrh > 0) then  
  podatak := mag(vrh)  
  vrh := vrh - 1  
endif
```

Magacin - upotreba

- Kod prevodjenja aritmetičkih izraza uz pomoć Poljske inverzne notacije.
- Kod poziva/povratka iz potprograma za:
 - prenos argumenata,
 - pamćenje adrese povratka i
 - pamćenje lokalnih promenljivih.

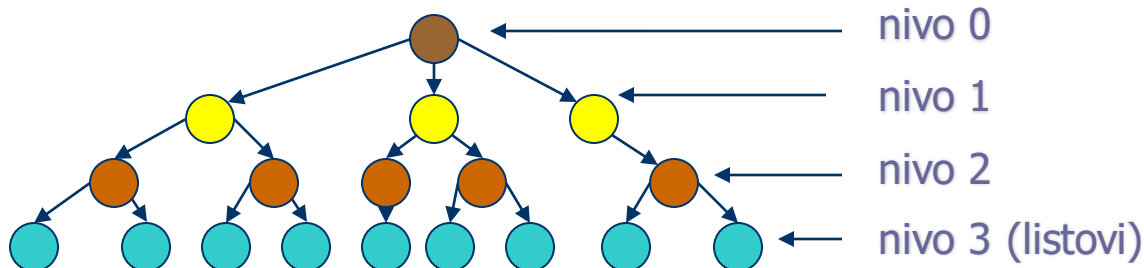
Red (FIFO – First-In-First-Out)

- Struktura podataka koja poštuje pravilo “Prvi-ušao-prvi izašao”:
 - Koristi se kada se elementi obraduju po redosledu pristizanja.
 - velika primena kod operativnih sistema: redovi čekanja procesa, red čekanja kod štampanja,...
- Upis se obavlja na jednom kraju (**rep**), a čitanje na drugom kraju (**čelo**, **vrh**).



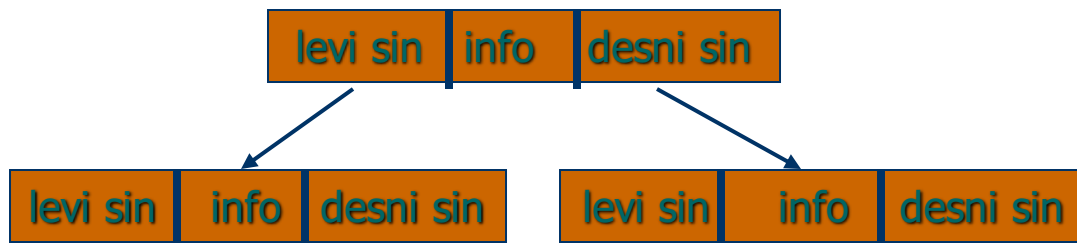
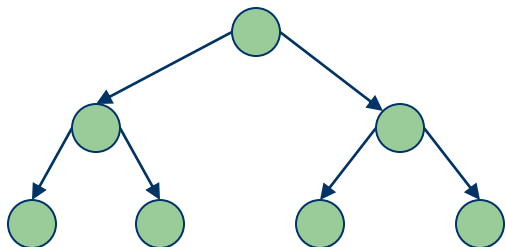
Nelinearne strukture – Stablo

- Svaki element strukture može imati veći broj prethodnika i sledbenika.
- Stablo predstavlja nelinearnu strukturu sa sledećim osobinama:
 1. Postoji jedan element koji se zove **koren stabla**. Na njega ne ukazuje ni jedan drugi element. Koren se nalazi na nivou 0.
 2. Na nivou 1 se nalaze elementi na koje ukazuje koren stable.
 3. Elementi nivoa 1 ukazuju na elemente nivoa 2,...
 4. Elementi nivoa $i-1$ ukazuju na elemente na nivou i .
 5. Elementi koji ne ukazuju na nove elemente čine **listove stabla**



Binarno stablo

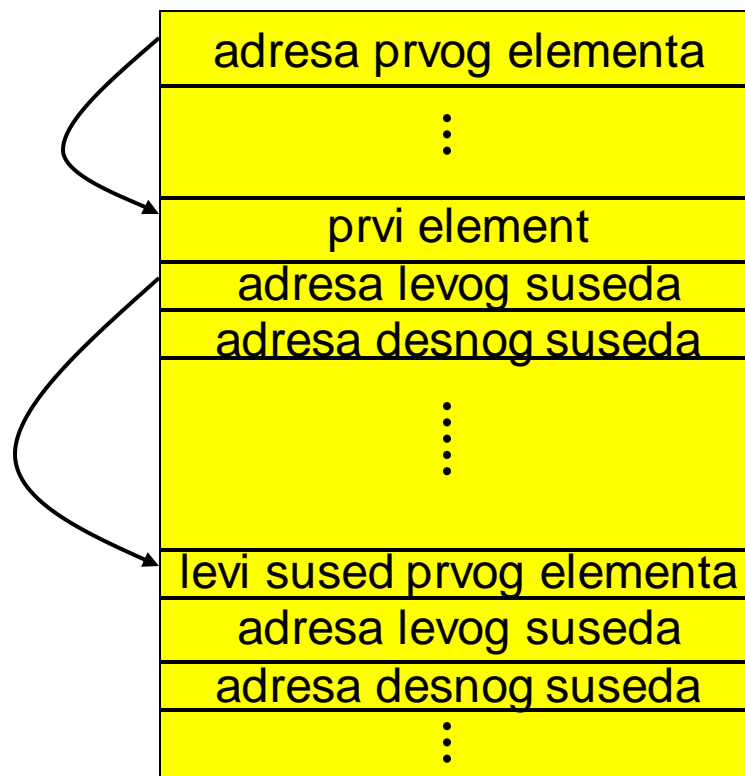
- Svaki čvor ima najviše dva sledbenika (potomka)



Karakteristike:

- nema petlji.
- izmedju svakog para čvorova postoji jedinstveni put.

Upotreba: kod pretraživanja sortiranih podataka.



Graf – struktura podataka

- Služe za predstavljanje matematičkih grafova.

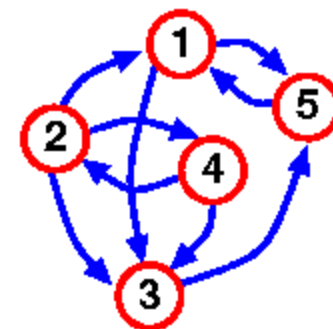
Čvorovi

- Preslikati čvorove u niz uzastopnih celih brojeva.
- Zapamtiti čvorove u polje.

Potezi (grane)

- **Matrica susedstva**
 - Logičke vrednosti -
TRUE (istina) – poteg postoji
FALSE (laž) - nema potega

- Upotreba: U računarskim mrežama,
putnim mrežama, PTT, kablovskoj televiziji,...



	1	2	3	4	5
1		X			X
2				X	
3	X	X		X	
4		X	X		
5	X		X		