

Strukture podataka i algoritmi

Prvo predavanje gradivni
elementi strukture podataka
LISTE

Tekstualni editor

 Novo Tekstni dokument - Blok za pisanje

Datoteka Uređivanje Oblikovanje Prikaz Pomoć

```
1 2 3 4 5
6 7 8 9 0
a b c d č
ć đ e f g
h i j k l
m n o p r
s š t u v
z ž _ - +
( ) / \ !
```

Heksa- decimalni editor

XVI32 - Novo Tekstni dokument.txt

File Edit Search Address Bookmarks Tools XVIscrip Help

File Edit Search Address Bookmarks Tools XVIscrip Help

0	31	20	32	20	33	20	34	20	35	0D	0A	1	2	3	4	5	□	□
B	36	20	37	20	38	20	39	20	30	0D	0A	6	7	8	9	0	□	□
16	61	20	62	20	63	20	64	20	E8	0D	0A	a	b	c	d	č	□	□
21	E6	20	F0	20	65	20	66	20	67	0D	0A	é	đ	e	f	g	□	□
2C	68	20	69	20	6A	20	6B	20	6C	0D	0A	h	i	j	k	l	□	□
37	6D	20	6E	20	6F	20	70	20	72	0D	0A	m	n	o	p	r	□	□
42	73	20	9A	20	74	20	75	20	76	0D	0A	s	š	t	u	v	□	□
4D	7A	20	9E	20	5F	20	2D	20	2B	0D	0A	z	ž	_	-	+	□	□
58	28	20	29	20	2F	20	5C	20	21	0D	0A	()	/	\	!	□	□
63	0D	0A										□	□					

Adr. dec: 0 Char dec: 49 Overwrite

Strukture podataka i algoritmi

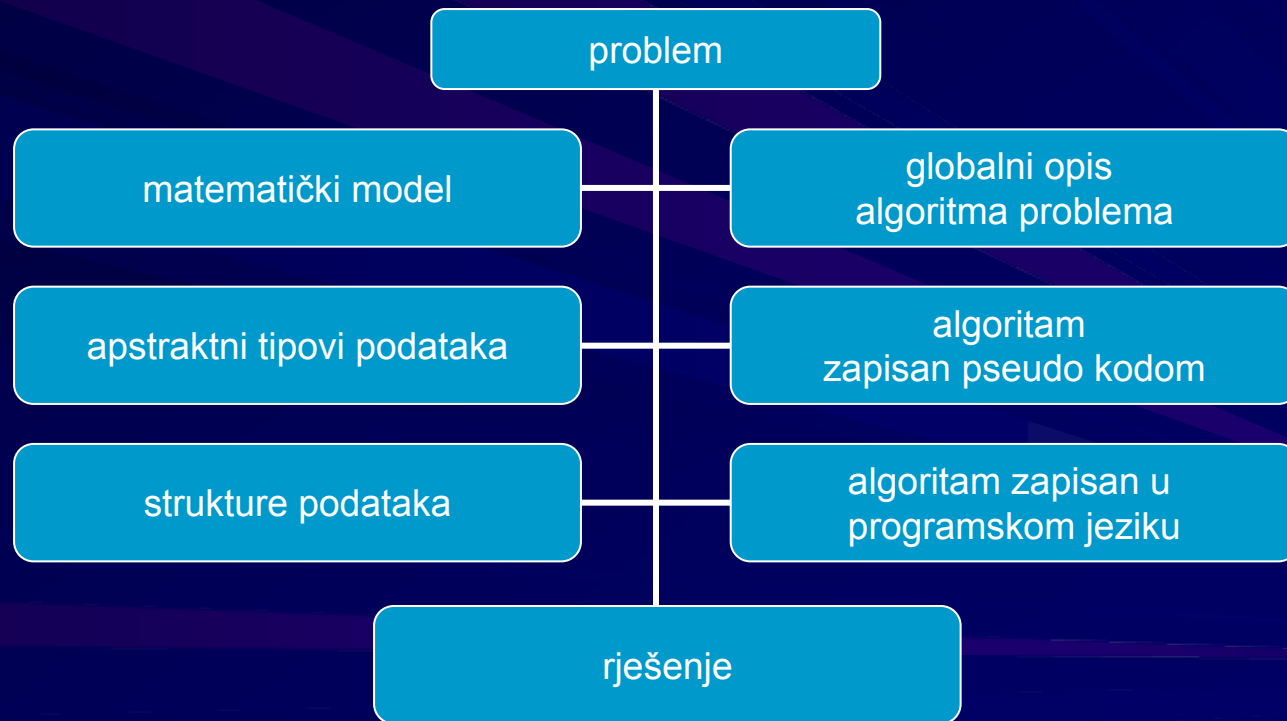
❖ Strukture podataka:

“statički aspekt programa”

❖ Algoritmi:

“dinamički aspekt programa”

Rješavanje problema metodom postepenog profinjavanja



Definicije pojmova (podaci)

❖ Tip podataka:

- *skup vrijednosti koje neki podatak može poprimiti (integer, real)*

❖ Apstraktni tip podataka

- *zadan jedan ili više tipova podataka, te jedna ili više operacija (procedura; funkcija)*
primjer kompleksni brojevi

❖ Struktura podataka

- *Skupina varijabli u nekom programu i veza među tim varijablama*

Algoritam

- ❖ Konkretna realizacija apstraktnog tipa podataka u nekom programu.
- Konačni niz instrukcija od kojih svaka ima jasno značenje i može biti izvršena u konačnom vremenu
- Iste instrukcije mogu se izvršiti više puta pod pretpostavkom da same instrukcije ukazuju na ponavljanje

Elementi od kojih se grade strukture podataka

- ❖ Klijetka (engl. cell)
- ❖ Polje
- ❖ Zapis
- ❖ Pokazivač (engl. pointer)
- ❖ Kursor

Klijetka (engl. cell)

- Varijabla koju promatramo kao zasebnu cjelinu
- Relativan pojam jer možemo analizirati i njenu unutarnju građu u određenim okolnostima
- Svaka klijetka ima tip i adresa
- Sadržaj klijetke odgovarajućeg tipa
- Gradivni element polja



Polje

- Mehanizam udruživanja manjih dijelova strukture u veće
- Više klijetki istog tipa pohranjeni na uzastopnim adresama
- Broj klijetki je unaprijed zadan i nepromjenljiv
- Indeksi cjelobrojne konstante

1

2

3

4

5

6



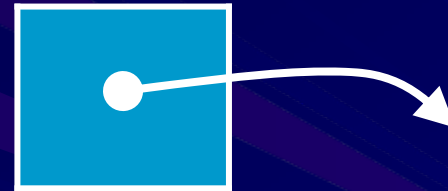
Zapis

- Još jedan način udruživanja manjih cjelina u veće strukture
- Zapis čini više klijetki koje ne moraju biti istog tipa ali su pohranjene na uzastopnim adresama
- Pojedina klijetka zove se komponenta zapisa
- Zapise možemo kombinirati u polja



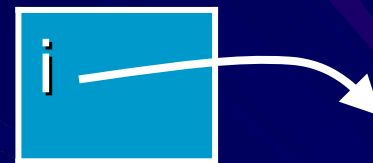
Pokazivač (engl. Pointer)

- Klijetka koja pokazuje na neku drugu klijetku
- Služi za uspostavljanje veze između dijelova strukture
- Sadržaj pokazivača je adresa klijetke koju treba pokazati

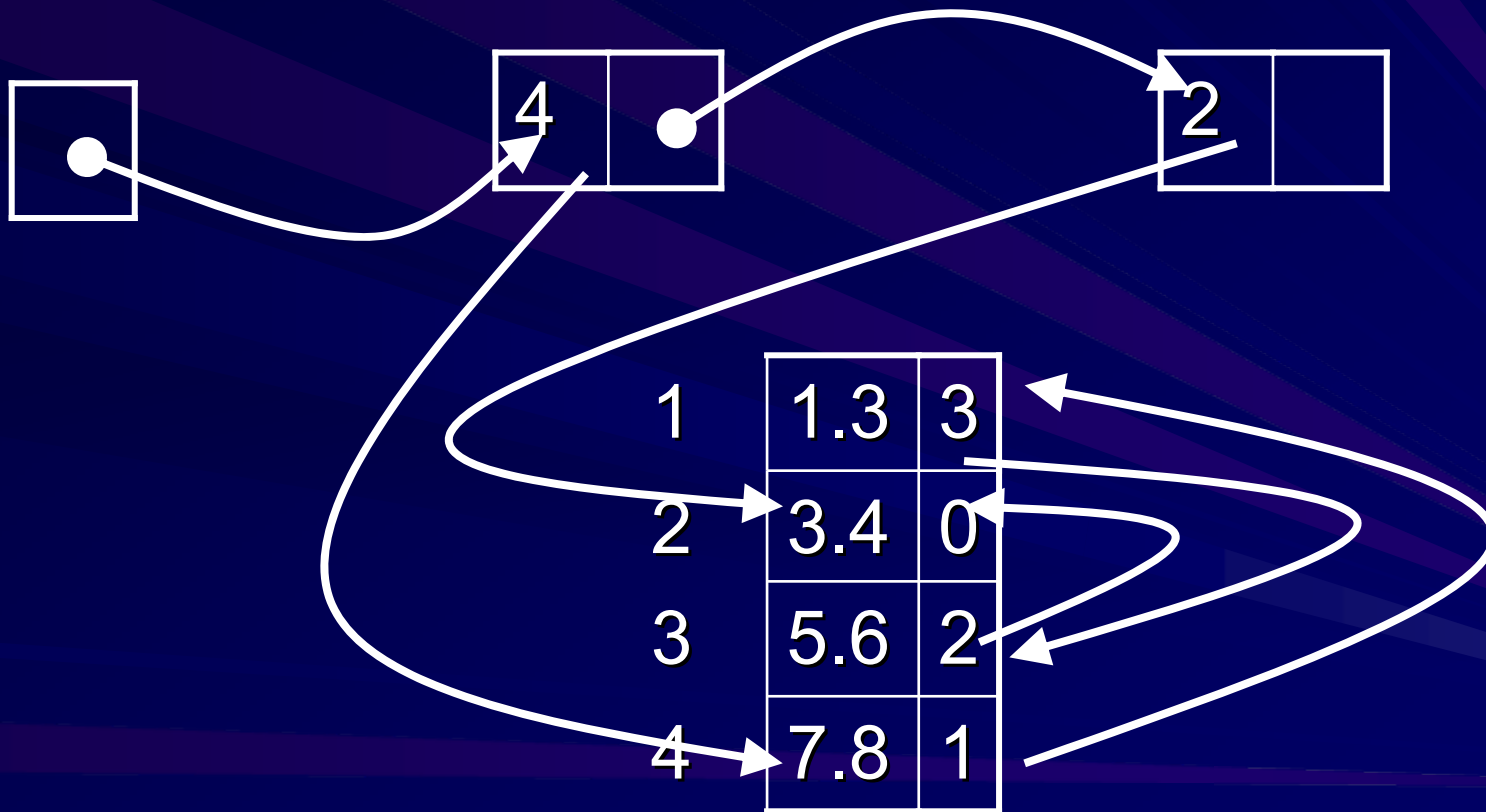


Kursor

- Također povezuje dijelove strukture
- Klijetka tipa integer koja pokazuje na element u polju
- Sadržaj kursora je indeks tog elementa polja



Primjer strukture podataka



Lista (općenito)

- ❖ Lista je konačan niz (od nule ili više podataka istog tipa. Podaci koji čine listu nazivaju se elementima liste
 - ✓ $[a_1, a_2, a_3, \dots, a_n]$
 - ✓ n - duljina liste
 - ✓ Ako je $n=0$ prazna lista
 - ✓ Definiran prethodnik i sljedbenik
 - ✓ Broj elemenata nije fiksiran
 - ✓ Identitet elementa liste određen njegovim pozicijom

Primjer liste

Polinom

$$P(x) = a_n x^{e_n} + a_{n-1} x^{e_{n-1}} + \dots + a_1 x^{e_1}$$

Gdje je $0 < e_1 < e_2 < \dots < e_n$

Zapravo se radi o listi

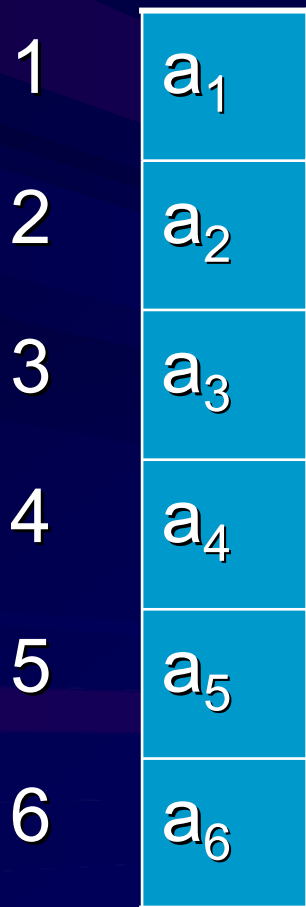
$$[(a_1, e_1), (a_2, e_2), \dots, (a_n, e_n)]$$

Operacije nad listama

- $\text{END}(L)$ – funkcija koja vraća poziciju na kraj liste
- $\text{MAKE_NULL}(L)$ – pretvara listu u praznu listu i vraća poziciju $\text{END}(L)$
- $\text{INSERT}(x,p,L)$ – ubacuje podatak x na poziciju p u listi L
- $\text{DELETE}(p,L)$ – izbacuje element p iz liste L
- $\text{FIRST}(L)$ – funkcija vraća prvu poziciju u listi, za praznu listu vraća $\text{END}(L)$
- $\text{NEXT}(p,L)$, $\text{PREVIOUS}(p,L)$ vraća poziciju ispred odnosno iza u listi
- $\text{RETRIVE}(p,L)$

Implementacija listi

■ Pomoću polja



■ Pomoću pokazivača

