ALGORITMI I STRUKTURE PODATAKA

RAČUNSKE VEŽBE – TERMIN BR. 3 – MATRICE I STRUKTURE

ALDINA AVDIĆ, DIPL. INŽ.

MATRICE

- **Primer 1.** Prikazati vektorsku prezentaciju retke matrice sa slike:

a) sa jednim vektorom zapisa od po tri polja,

b) sa tri posebna vektora.

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 8 & 10 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 & 0 \end{bmatrix}$$

MATRICE

A) Svaki zapis se sastoji od

- broja vrste (R - row)
- broja kolone (C - column)
- vrednosti (V - value)

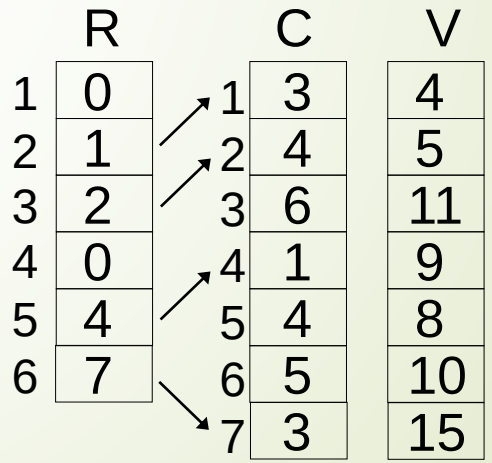
R	C	V
2	3	4
3	4	5
3	6	11
5	1	9
5	4	8
5	5	10
6	3	15

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 8 & 10 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 & 0 \end{bmatrix}$$

MATRICE

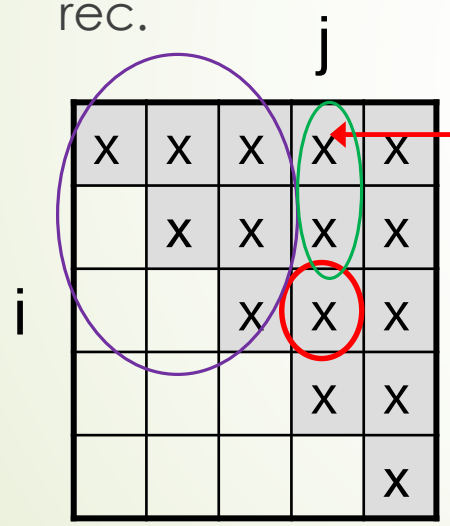
- B) Potrebno je formirati 3 nezavisna vektora
  - Vektor V sadrži vrednosti i ima onoliko elemenata koliko ima nenultih elemenata matrice
  - Vektor C sadrži broj kolone odgovarajućeg elementa vektora V
  - Vektor R sadrži indeks prvog elementa vektora C koji odgovara datom elementu vektora R i ima onoliko elemenata koliko i vrsta matrice

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 8 & 10 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 & 0 \end{bmatrix}$$



MATRICE

- **Primer 2.** Objasniti postupak smeštanja i izvesti adresnu funkciju pri pristupu proizvoljnom elementu gornje trougaone matrice **smeštene po kolonama**. Smatrati da se jedan element matrice smešta u tačno jednu memorijsku reč.



Broj elemenata
pre prvog elementa
kolone j :

$$\sum_{k=1}^{j-1} k = \frac{j(j-1)}{2}$$

Broj elemenata
u koloni j
pre traženog elementa:

$$i-1$$

$$A_{ij} = A_{11} + (j(j-1)/2 + i - 1)*s$$

MATRICE

Primer 3. Tridijagonalna matrica je matrica reda nxn, gde je A[i,j] = 0, ako je |i-j|>1.

a) Koliki je maksimalan broj nenultih elemenata ?

b) Ako se matrica linearizuje po vrstama izvesti adresnu funkciju.

a_{11}	a_{12}			
a_{21}	a_{22}	a_{23}		
	a_{32}	a_{33}	a_{34}	
		a_{43}	a_{44}	a_{45}
			a_{54}	a_{55}

a) Max. broj nenultih elemenata: $n_{nz} = 3*n - 2$

b) Matrica se smešta po vrstama:

a_{11}	a_{12}	a_{21}	a_{22}	a_{23}	a_{32}	a_{33}	...
0	1	2	3	4	5	6	...

$A_{ij} = A_{11} + (3(i-1) + j - i + 1 - 1)*s$, ako je $|i-j| <= 1$

$A_{ij} = A_{11} + (2*i + j - 3)*s$, ako je $|i-j| <= 1$

STRUKTURE (SLOGOVI, REKORDI)

- struct [structure tag] {
- member definition;
- member definition;
- ...
- member definition;
- } [one or more structure variables];

- struct Books {
- char title[50];
- char author[50];
- char subject[100];
- int book_id;
- } book;

STRUKTURE – PRISTUP ELEMENTIMA

```
int main( ) {

   struct Books Book1;        /* Declare Book1 of type Book */
   struct Books Book2;        /* Declare Book2 of type Book */

   /* book 1 specification */
   strcpy( Book1.title, "C Programming");
   strcpy( Book1.author, "Nuha Ali");
   strcpy( Book1.subject, "C Programming Tutorial");
   Book1.book_id = 6495407;

   /* book 2 specification */
   strcpy( Book2.title, "Telecom Billing");
   strcpy( Book2.author, "Zara Ali");
   strcpy( Book2.subject, "Telecom Billing Tutorial");
   Book2.book_id = 6495700;

   /* print Book1 info */
   printf( "Book 1 title : %s\n", Book1.title);
   printf( "Book 1 author : %s\n", Book1.author);
   printf( "Book 1 subject : %s\n", Book1.subject);
   printf( "Book 1 book_id : %d\n", Book1.book_id);

   /* print Book2 info */
   printf( "Book 2 title : %s\n", Book2.title);
   printf( "Book 2 author : %s\n", Book2.author);
   printf( "Book 2 subject : %s\n", Book2.subject);
   printf( "Book 2 book_id : %d\n", Book2.book_id);

   return 0;
}
```

STRUKTURE - POTPROGRAMI

```
   /* print Book1 info */
   printBook( Book1 );

   /* Print Book2 info */
   printBook( Book2 );

   return 0;
}

void printBook( struct Books book ) {

   printf( "Book title : %s\n", book.title);
   printf( "Book author : %s\n", book.author);
   printf( "Book subject : %s\n", book.subject);
   printf( "Book book_id : %d\n", book.book_id);
}
```

STRUKTURE I POKAZIVAČI

```
   /* print Book1 info by passing address of Book1 */
   printBook( &Book1 );

   /* print Book2 info by passing address of Book2 */
   printBook( &Book2 );

   return 0;
}

void printBook( struct Books *book ) {

   printf( "Book title : %s\n", book->title);
   printf( "Book author : %s\n", book->author);
   printf( "Book subject : %s\n", book->subject);
   printf( "Book book_id : %d\n", book->book_id);
}
```

TYPEDEF

```c
#include <stdio.h>
#include <string.h>

typedef struct Books {
   char title[50];
   char author[50];
   char subject[100];
   int book_id;
} Book;

int main( ) {

   Book book;

   strcpy( book.title, "C Programming");
   strcpy( book.author, "Nuha Ali");
   strcpy( book.subject, "C Programming Tutorial");
   book.book_id = 6495407;

   printf( "Book title : %s\n", book.title);
   printf( "Book author : %s\n", book.author);
   printf( "Book subject : %s\n", book.subject);
   printf( "Book book_id : %d\n", book.book_id);

   return 0;
}
```

NIZ STRUKTURA

```c
#include <stdio.h>
#include <string.h>

struct student
{
     int id;
     char name[30];
     float percentage;
};

int main()
{
     int i;
     struct student record[2];

     // 1st student's record
     record[0].id=1;
     strcpy(record[0].name, "Raju");
     record[0].percentage = 86.5;

     // 2nd student's record
     record[1].id=2;
     strcpy(record[1].name, "Surendren");
     record[1].percentage = 90.5;

     // 3rd student's record
     record[2].id=3;
     strcpy(record[2].name, "Thiyagu");
     record[2].percentage = 81.5;

     for(i=0; i<3; i++)
     {
          printf("     Records of STUDENT : %d \n", i+1);
          printf(" Id is: %d \n", record[i].id);
          printf(" Name is: %s \n", record[i].name);
          printf(" Percentage is: %f\n\n",record[i].percentage);
     }
     return 0;
}
```

Hvala na pažnji!