

Grafuri neorientate

Structuri de date neliniare.

Reprezentarea lor:

- matricea de adiacență

 $a_{ij} = 1$ dacă (i, j) muchie $a_{ij} = 0$ dacă (i, j) nu este muchie $n =$ nr de noduri

$$A = (a_{ij})_{n \times n}$$

0	1	0	0	0
1	0	1	0	0
0	1	0	1	0
0	0	1	0	1
0	0	0	1	0

info = vector
de n elem.info[i] = info
nodului i $a_{ij} = a_{ji}$ matrice simetrică graf. neorientat

Operații de bază:

1. Accesarea

: un nod info[i]

: o muchie

if $a[i][j] = 1$ thenwrite (i, j) muchie;

2. Inserare

a) A unei muchii

Se dau i, j if $1 \leq i \leq n$ and $1 \leq j \leq n$ thenif $a[i][j] = 0$ then $a[i][j] = 1; a[j][i] = 1;$

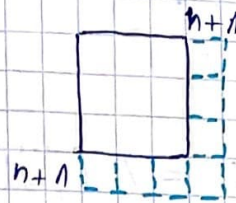
else write "muchie existenta"

endif

else write "indici eronati"

endif

b) a unui nod;
citim informația nodului (a)
info $[n+1] = a$.



```
for i = 1, n
  if (i, n+1) muchie then
    a[i][n+1] = 1
    a[n+1][i] = 1
  else a[i][n+1] = 0
       a[n+1][i] = 0
  endif
endfor
n++;
```

3. a) Ștergerea unei muchii

Se dau i, j

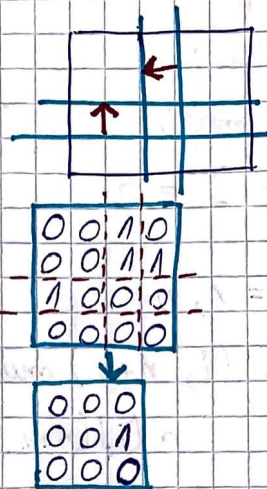
```
if 1 ≤ i ≤ n and 1 ≤ j ≤ n then
  if a[i][j] = 1 then
    a[i][j] = 0
    a[j][i] = 0
  else write '(i, j) nu este muchie'
endif
```

b) Ștergerea unui nod: i

```
if not (1 ≤ i ≤ n) then
  write 'nod inexistent'
  STOP
endif
```


col. i col. i+1
 $1 \text{ } C[i] \leftarrow C[i+1]$

for $j = 1, n$
 $A[j][i] = A[j][i+1]$
endfor



2 $col[i+1] \leftarrow col[i+2]$
for $j = 1, n$
 $A[j][i+1] = A[j][i+2]$
endfor

3 $... col[n-1] \leftarrow col[n]$

! Stergere coloana i

for $k = i, n-1$ mutăm coloanute spre stânga
coloanute $i+1, \dots, n$
for $j = 1, n$
 $A[j][k] = A[j][k+1]$
endfor
endfor

3 Stergem linia i : mutăm în sus liniile $i+1, \dots, n$

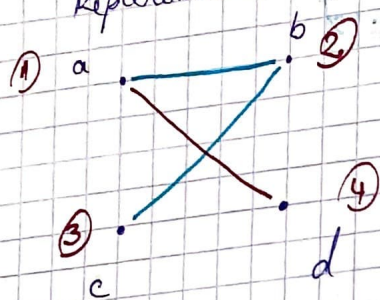
for $k = i, n-1$
for $j = 1, n$
 $A[k][j] = A[k+1][j]$
endfor
endfor

info "Alin" "Mihai" "George" "Irina"

for $j = i, n-1$
info[j] = info[j+1]
endfor
n--;

Reprezentare cu liste de adiacență.

info	a	b	c	d
	1	2	3	4



Fiecare nod va avea o listă în care trecem doar nodurile adiacente, în ordine creșterea.

Adj			
1	1	→	2 → NULL
2	2	→	1 → 3 → NULL
3	3	→	2 → NULL
4	4	→	NULL

$Adj[i] = \text{lista corespunzătoare nodului } i$

$Nod * Adj[10]$

// maxim 10

$n = \text{nr. de noduri existente}$

Struct Nod

```
{
    int info;
    Nod * link;
}
```

```
T INFO[10];
Nod * Adj[10];
```

Operații de bază:

1. Accesarea elementelor

- nodul i : $info[i]$

- toate muchiile nodului i // parcurgem lista de adiacență a lui i

$iter = Adj[i]$ while $iter \neq NULL$

while write

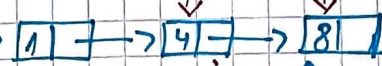
write 'muchie: ' i , $iter \rightarrow info$

$iter = iter \rightarrow link$

end while

if $Adj[i] == NULL$ then

write "nod izolat"

2. Inserarea unei muchii (i, j) $i \rightarrow$ 



$p = \text{new Nod};$

if $p = \text{NULL}$ then write 'overflow'
endif

$p \rightarrow \text{info} = j;$

$p \rightarrow \text{link} = \text{NULL};$

$\text{Adj}[i] = p;$

lista nevida:

$p = \text{new Nod}$ then write "OVERFLOW"

if $(p == \text{NULL})$ then
write "overflow"
STOP

$p \rightarrow \text{info} = j;$

while $\text{iter} \neq \text{NULL}$ and $\text{iter} \rightarrow \text{info} < j$
ante = iter;
iter = iter \rightarrow link

if $\text{iter} = \text{NULL}$ then

ante \rightarrow link = p ;

$p \rightarrow \text{link} = \text{NULL};$

else if $\text{iter} \rightarrow \text{info} = j$ then write "muchie existenta"

else "iter \rightarrow info $> j$ "

$p \rightarrow \text{link} = \text{iter}$

ante \rightarrow link = p
endif