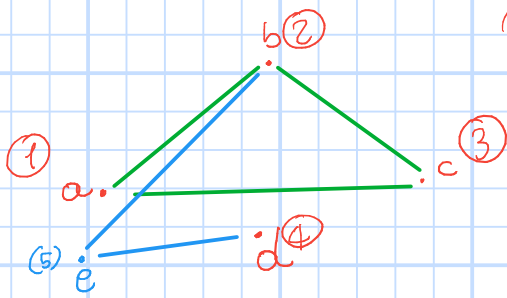


# Grafuri - Reprezentare cu liste de adiacență

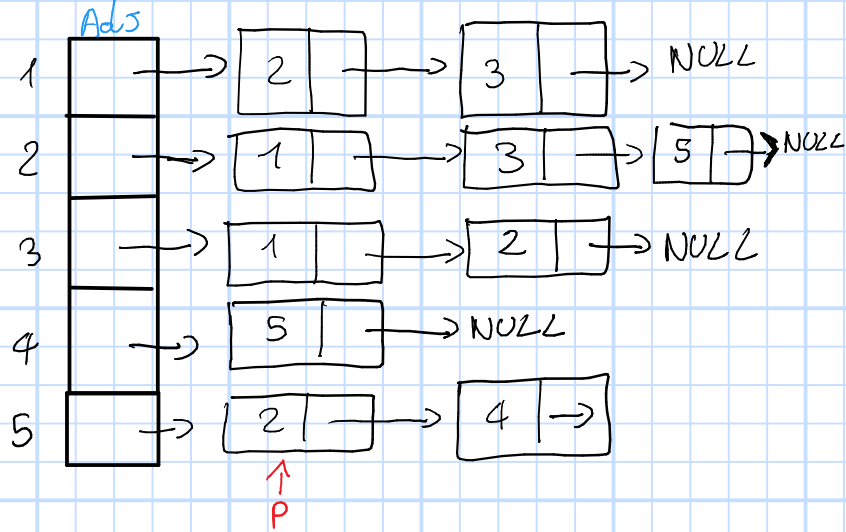
G = GRAF NEORIENTAT!

$n = nr. \text{ noduri}$  | Pentru fiecare nod, o listă de noduri cu care nodul este adiacent

Listă se reprezintă printr-o listă simplu enlantuată în care nodurile se trec în ordine crescătoare



$n=4$



struct Node {

int info;  
Node \* link;

};

$N = nr. \text{ maxim de noduri}$

Node \* Adj[N];

## OPERATII DE BAZA

- Accessare → muchie  
→ Noduri
- Inserare → muchie  
→ Noduri

Nodul nou:  $n+1$

Se inserează împreună cu toate  $n$  ile

Adj[n+1] = NULL; ante = NULL;

For  $i=1, n$   
if (n+1, i) muchie then // adăugăm nodul  $[i] \rightarrow NULL$  în listă  
p = next Node;  
if p = NULL then stop;  
endif  
p → info = i  
p → link = NULL  
if Adj[n+1] = NULL then Adj[n+1] = p  
else ante → link = p  
endif  
ante = p;



// adaugă  $n+1$  în lista modulară  $i$

```

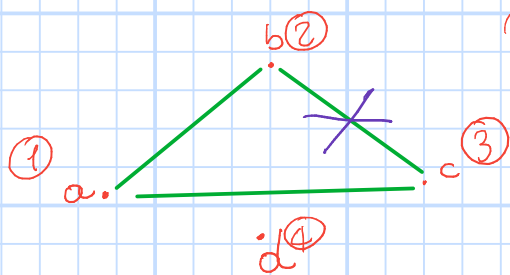
iter = Ads[i]
if Ads[i] = NULL then // dacă lista lui i este vidă
    p = New Node;
    if p = NULL then write "OVERFLOW", stop;
    [endif]
    p->info = n+1;
    p->link = NULL;
    Ads[i] = p;
else // dacă lista lui i nu este vidă
    iter = Ads[i]
    while iter->link ≠ NULL
        iter = iter->link
    [endwhile]
    iter->link = p;
[endif]
[endif]

```

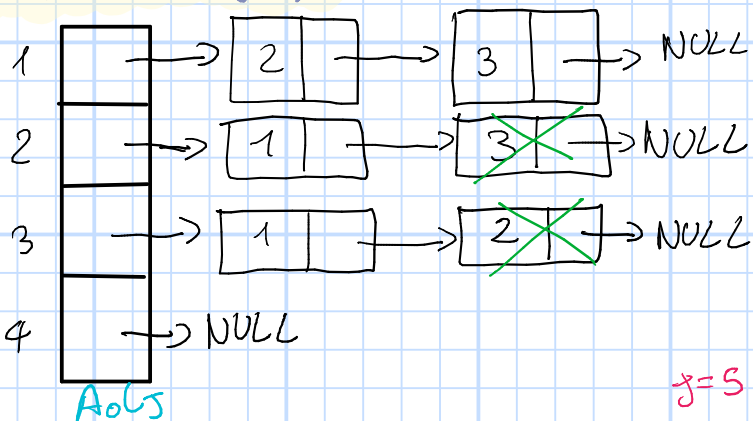
endifor.  
 $n+1$ ;

stergere a unei muchii:  $(i, j)$

elimina-muchie  $(i, j)$ :

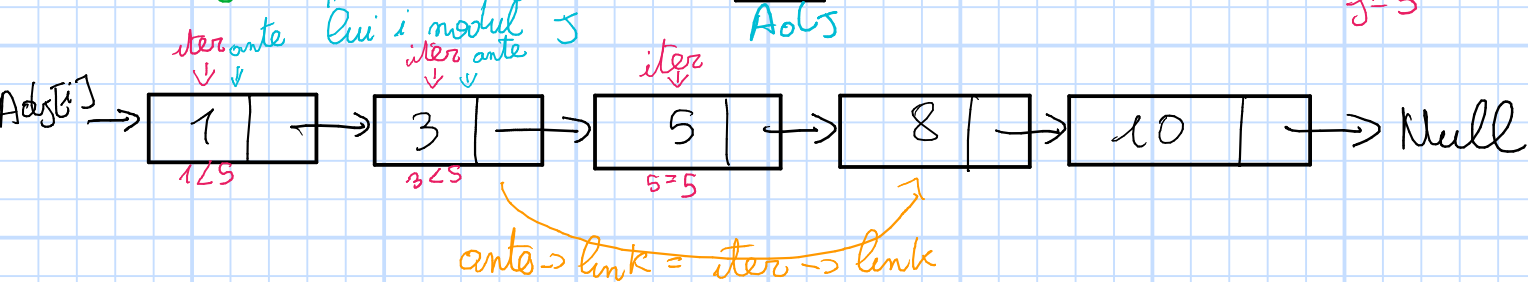


$n=4$



$j=5$

eliminăm  $(2, 3)$  // eliminăm din lista



```

if !(1 ≤ i ≤ n and 1 ≤ j ≤ n) then write "indici incorecti", stop;
[endif]

```

```

if Ads[i] = Null then write "Muchia nu există" // listă vidă
    Stop;
[endif]

```

```

iter = Ads[i], onto = Null;
while iter ≠ Null && iter->info < j
    onto = iter;
    iter = iter->link;
[endwhile]

```



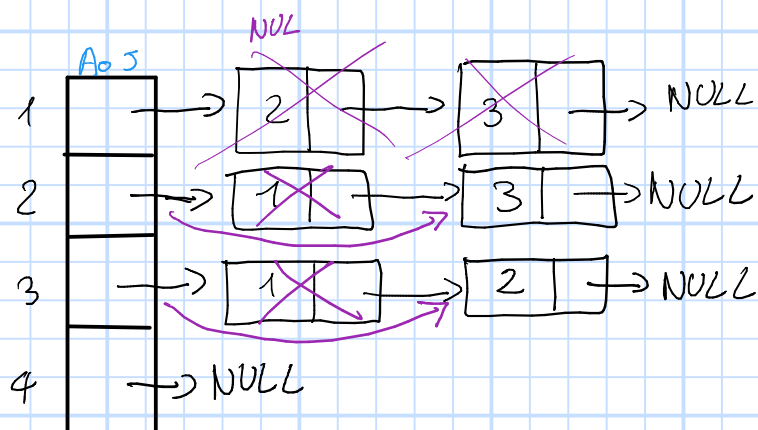
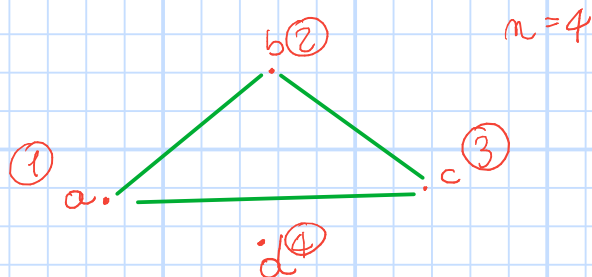
```

if iter  $\neq$  NULL then
    if iter  $\rightarrow$  info = 5
        ante  $\rightarrow$  link = iter  $\rightarrow$  link;
        delete iter;
    else // iter  $\rightarrow$  info > 5
        write "muchia (i, 5) nu exista"
    endif
else
    write "muchia (i, 5) nu exista";
endif

```

Pentru eliminarea completă a muchiei  $(i, 5)$ , vom apela  $\text{elimina\_muchie}(i, 5)$  | GRAF  
 $\text{elimina\_muchie}(5, i)$  | NEOR.

Stergerea unui nod



Fie  $i$  nodul care trebuie sters

Dacă  $\text{Adj}[i] = \text{NULL}$  then lista lui este vidă dar renumerotăm nodurile,  $n--$

Dacă  $i = n$  atunci doar  $n--$ ;

Eliminăm nodul  $i$  |  $\text{elimina\_nod}(i)$ ;

```

if  $!(1 \leq i \leq n)$  write "indice invalid"
stop

```

```

endif

```

```

if  $\text{Adj}[i] \neq \text{NULL}$  then
    iter =  $\text{Adj}[i]$ ;

```



```

while iter  $\neq$  NULL
    j = iter  $\rightarrow$  info;
    // eliminam din lista lui j nodul i
    call elimina_nodul(j, i);
    // iar nodurile urmatoare din lista trebuie sa primeasca (info--)
    call renumerotare(j, i); // renumerotarea din lista j cu valori > i
    iter = iter  $\rightarrow$  link;
endwhile

```

// Curatam lista lui i

```
call deleteLista(i);
```

```
endif // renumerotam nodurile
```

```
For j = i, m
```

```
    INFO[j] = INFO[j+1];
```

```
endfor
```

```
m--;
```

```
renumerotare(i, j)
```

```
if Ad[j] = NULL
```

```
    write "Nu avem ce sterge"
    stop;
endif
```

```
iter = Ad[j]
```

```
while iter  $\neq$  NULL
```

```
    if iter  $\rightarrow$  info > i
        iter  $\rightarrow$  info--;
```

```
    endif
```

```
    iter = iter  $\rightarrow$  link;
```

```
endwhile
```

```
deleteLista(i)
```

```
if Ad[j] = NULL
```

```
    write "Nu avem ce sterge"
    stop;
endif
```

```
iter = Ad[i]
```

```
while iter  $\neq$  NULL
```

```
    temp = iter;
```

```
    Ad[i] = iter  $\rightarrow$  link;
```

```
    delete temp;
```

```
    iter = Ad[i];
```

```
endwhile
```