

# Algoritmica grafurilor

## II. Reprezentări, parcurgeri în grafuri, drumuri

Mihai Suciu

Facultatea de Matematică și Informatică (UBB)  
Departamentul de Informatică

Martie, 4, 2025



## ① Reprezentarea / memorarea grafurilor

- Lista de adiacență
- Matricea de adiacență
- Matricea de incidentă
- Exemple

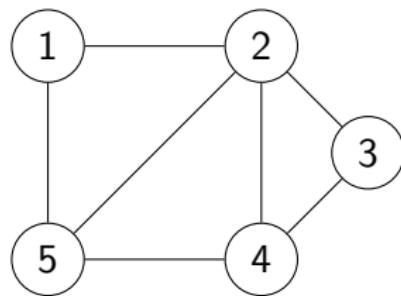
## ② Paarcurgeri în lățime și adâncime

- Parcursere în lățime
- Parcursere în adâncime
- Exemple
- Kosaraju



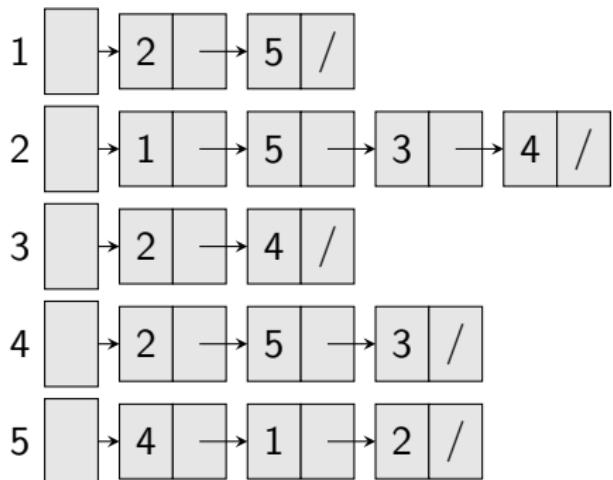
## Exemplu - graf neorientat

graf → listă adiacență → matrice de adiacență





# Lista de adiacență și matricea de adiacență

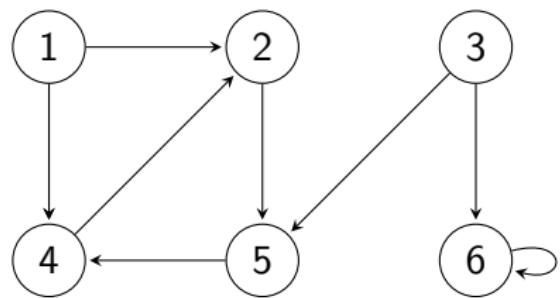


$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$



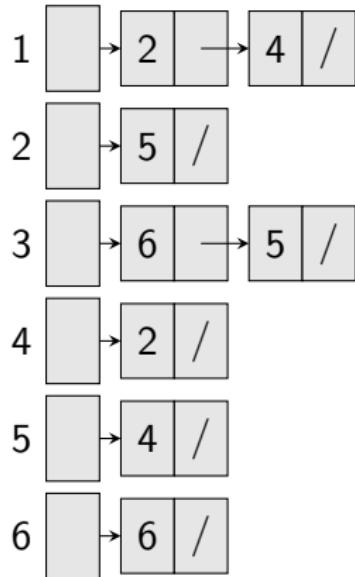
## Exemplu - graf orientat

graf → listă adiacență → matrice de adiacență





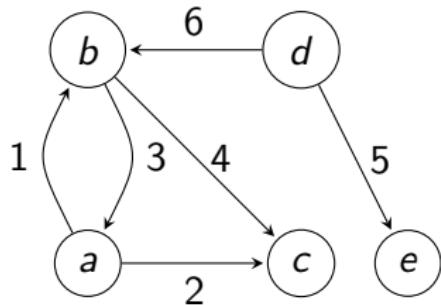
# Lista de adiacență și matricea de adiacență



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



## Exemplu - graf orientat, matricea de incidentă



$$B = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$



# Parcurgere în lățime (BFS) - procedura

BFS( $G, s$ )

```
1: for fiecare vârf  $u \in G, V - \{s\}$  do
2:    $u.\text{color} = \text{alb}$ 
3:    $u.d = \infty$ 
4:    $u.\pi = \text{NIL}$ 
5: end for
6:  $s.\text{color} = \text{gri}$ 
7:  $s.d = 0$ 
8:  $s.\pi = \text{NIL}$ 
9:  $Q = \emptyset$ 
10: Enqueue(Q,s)
11: while  $Q \neq \emptyset$  do
12:    $u = \text{Dequeue}(Q)$ 
13:   for fiecare  $v \in G.\text{Adj}[u]$  do
14:     if  $v.\text{color} == \text{alb}$  then
15:        $v.\text{color} = \text{gri}$ 
16:        $v.d = u.d + 1$ 
17:        $v.\pi = u$ 
18:       Enqueue(Q,v)
19:     end if
20:   end for
21:    $u.\text{color} = \text{negrui}$ 
22: end while
```

# BFS



- durata în timp a algoritmului este  $O(V + E)$

Exemplu

# Parcugere în adâncime (DFS) - procedura



DFS( $G$ )

```
1: for fiecare vârf  $u \in G.V$  do
2:   u.color = alb
3:   u. $\pi$  = NIL
4: end for
5: time = 0
6: for fiecare  $u \in G.V$  do
7:   if u.color == alb then
8:     DFS_VISIT( $G, u$ )
9:   end if
10: end for
```



## DFS - procedura (II)

DFS\_VISIT( $G, u$ )

```
1: time = time + 1
2:  $u.d = time$ 
3:  $u.color = \text{gri}$ 
4: for fiecare  $v \in G.Adj[u]$  do
5:   if  $v.color == \text{alb}$  then
6:      $v.\pi = u$ 
7:     DFS_VISIT( $G, v$ )
8:   end if
9: end for
10:  $u.color = \text{negru}$ 
11: time = time + 1
12:  $u.f = time$ 
```

# Exemple

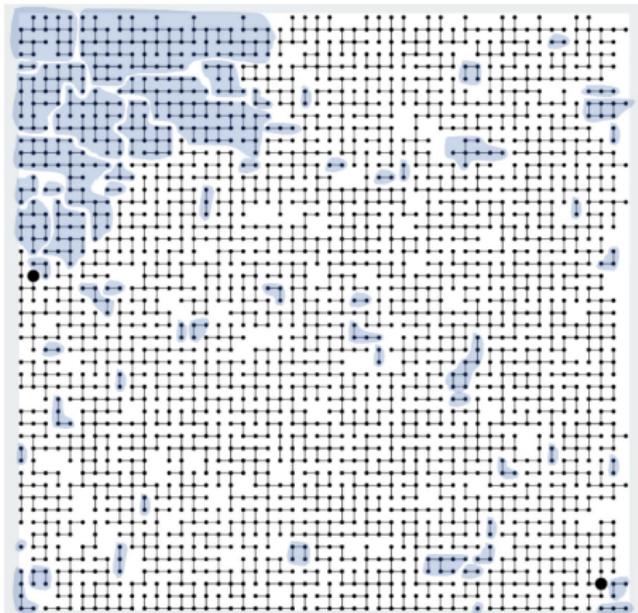


- Relația: Din orice punct puteți ajunge la orice punct
- Câte componente conexe are următorul graf?



## Exemple

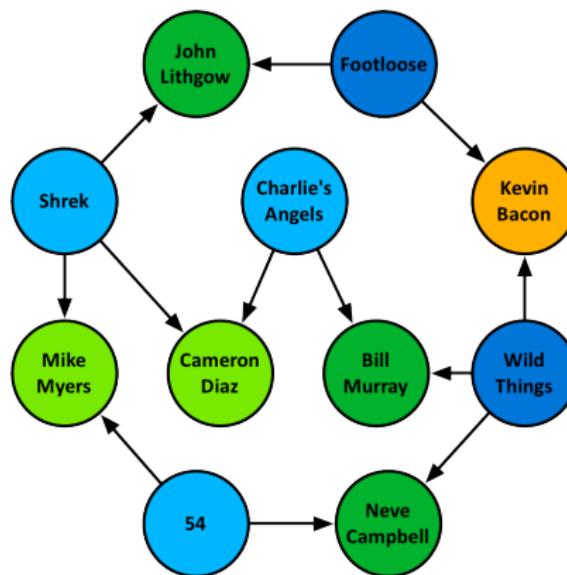
- Relația: Din orice punct puteți ajunge la orice punct
- Câte componente conexe are următorul graf?





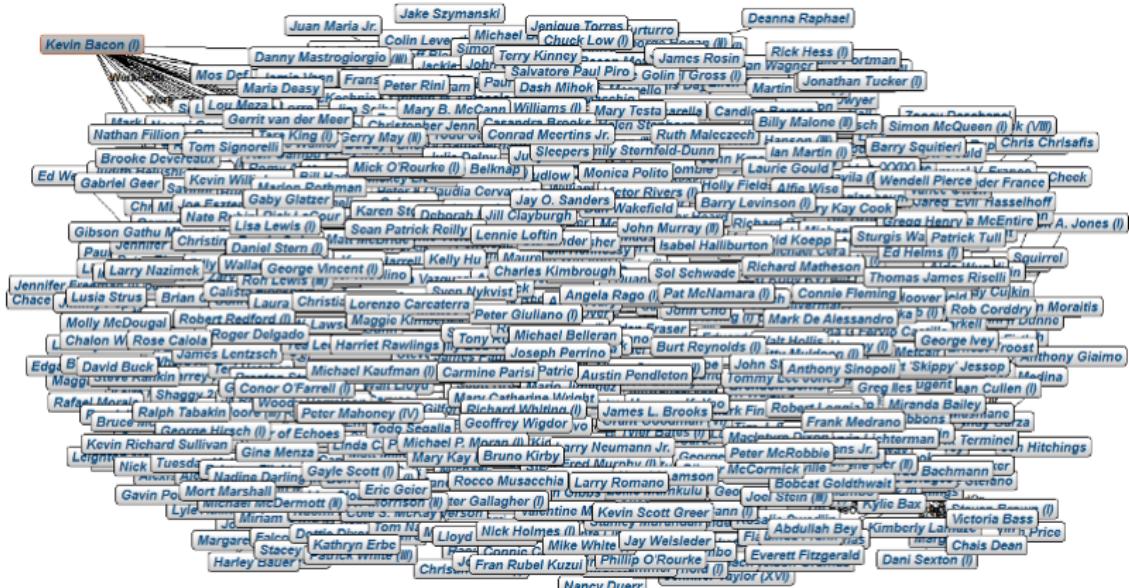
## Exemple (II)

- facebook - sugestie de noi prieteni pe baza BFS
- numărul Kevin Bacon / Erdős Pál





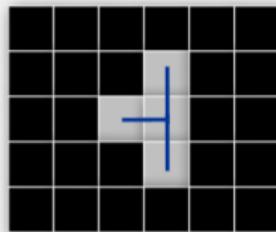
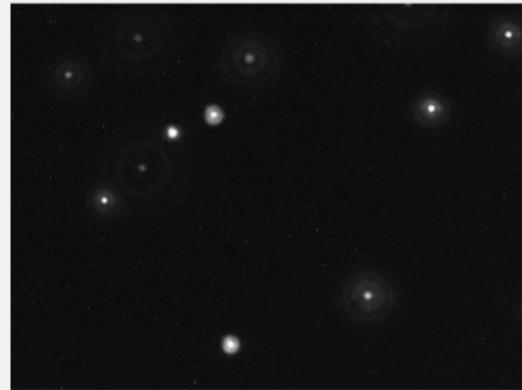
# Exemple (II)





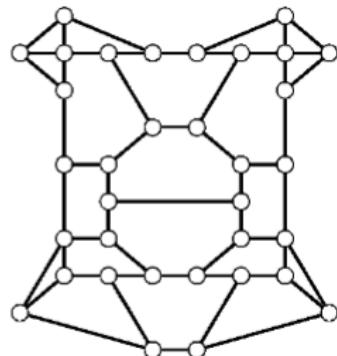
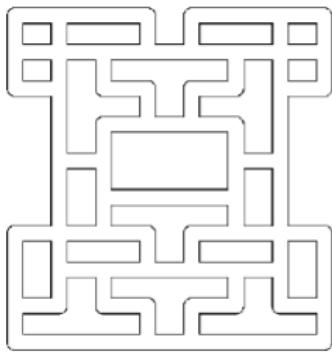
## Exemple (III) - prelucrare de imagini

- să se caute stelele mai mari din imagine





## Exemple (IV) - parcugerea unui labirint

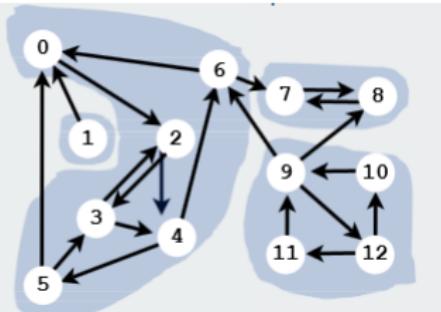
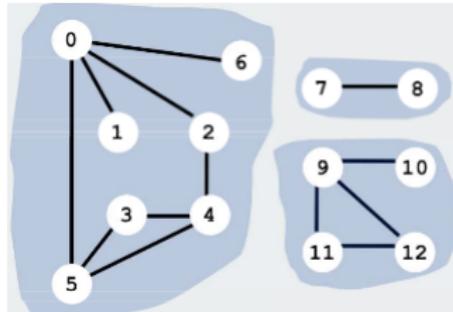


Algoritmul lui Thremaux - secolul 19, bazat pe DFS



# Graf tare conex, slab conex - exemplu

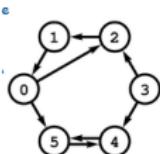
- componente conexe pe grafuri orientate / neorientate (DFS)



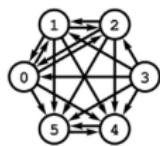


# Exemplu DFS

Închiderea tranzitivă a unui graf



	0	1	2	3	4	5
0	1	0	1	0	0	1
1	1	1	0	0	0	0
2	0	1	1	0	0	0
3	0	0	1	1	1	0
4	0	0	0	0	1	1
5	0	0	0	0	1	1



	0	1	2	3	4	5
0	1	1	1	0	1	1
1	1	1	1	0	1	1
2	1	1	1	0	1	1
3	1	1	1	1	1	1
4	0	0	0	0	1	1
5	0	0	0	0	1	1



# Algoritmul Kosaraju - Sharir

- algoritm pentru determinarea componentelor tare conex dintr-un graf orientat
- pași
  - DFS cu vârfurile puse pe o stivă
  - DFS pe complementul grafului

Exemplu



# Cel mai scurt drum / lanț

- pentru un graf neponderat, orientat sau neorientat, putem folosi algoritmul lui Moore pentru a găsi cel mai scurt drum / lanț
- notații
  - $u$  - nodul sursă
  - $l(v)$  - lungimea drumului
  - $p(v)$  - părintele vârfului  $v$
  - $Q$  - o coadă



# Algoritmul lui Moore

MOORE( $G, u$ )

1.  $I(u) := 0$
2. **for** toate vârfurile  $v \in V(G)$ ,  $v \neq u$  **do**
  3.  $I(v) := \infty$
  4.  $Q = \emptyset$
  5.  $u \rightarrow Q$
  6. **while**  $Q \neq \emptyset$  **do**
    7.  $Q \rightarrow x$
    8. **for** toți vecinii  $y \in N(x)$  **do**
      9. **if**  $I(y) = \infty$  **then**
        10.  $p(y) := x$
        11.  $I(y) := I(x) + 1$
        12.  $y \rightarrow Q$
    13. **return**  $I, p$



## Algoritmul lui Moore (II)

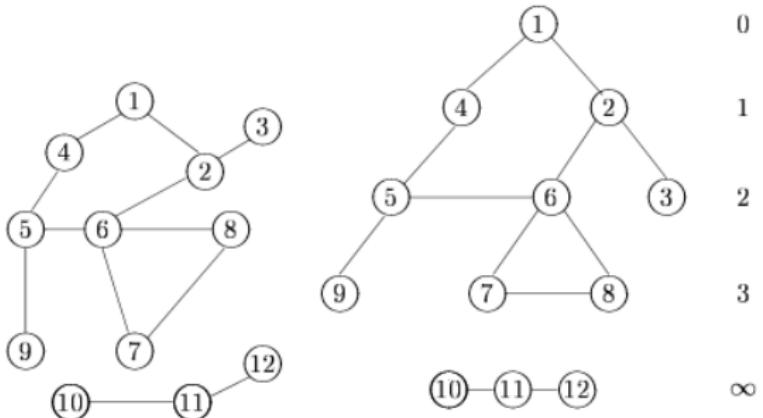
- știind  $l, p, v$  cum putem afla drumul

MOORE\_DRUM( $l, p, v$ )

1.  $k := l(v)$
2.  $u_k := v$
3. **while**  $k \neq 0$  **do**
4.      $u_{k-1} := p(u_k)$
5.      $k := k - 1$
6. **return**  $u$



# Exemplu



	1	2	3	4	5	6	7	8	9	10	11	12
$l$	0	1	2	1	2	2	3	3	3	$\infty$	$\infty$	$\infty$
$p$	1	2	1	4	2	6	6	5				