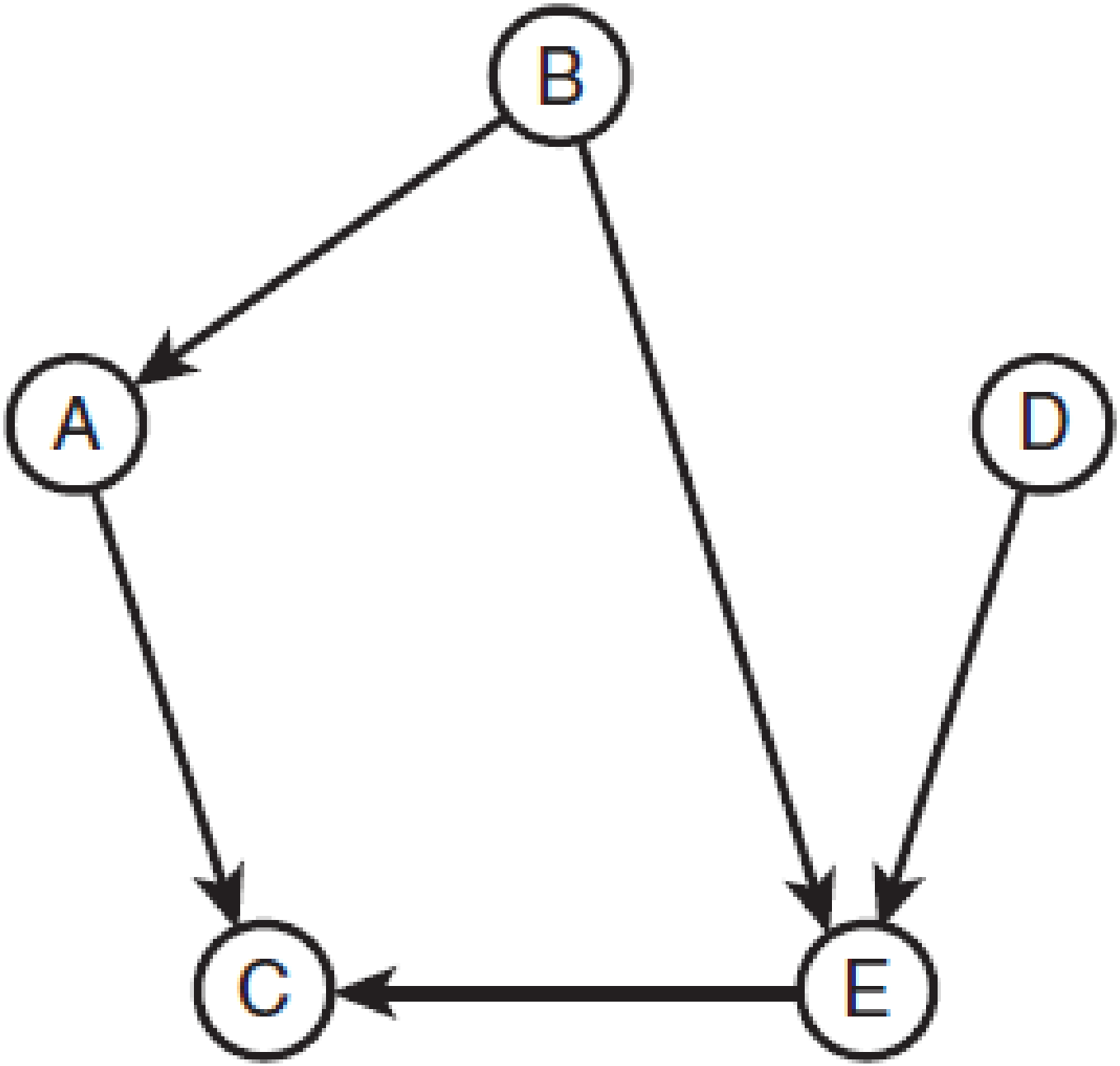
Universitatea Politehnica din București
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare

ALGORITMUL LUI WARSHALL

Conectivitate în grafuri orientate

- Într-un graf neorientat, se pot găsi toate vârfurile care sunt conectate, utilizând parcurgerea în adâncime sau parcurgerea pe nivel
- Dacă se dorește găsirea tuturor vârfurilor conectate într-un graf orientat, nu se poate porni dintr-un vârf selectat aleator, pentru a ajunge la toate celelalte vârfuri conectate


B
A
D
C
E

Exemplu

- Dacă se pornește din A, se poate ajunge la C, dar nu și la alte vârfuri
- Dacă se pornește din B, nu se poate ajunge la D
- Dacă se pornește din C, nu se poate ajunge la niciun alt vârf
- La ce vârfuri se poate ajunge dacă se pornește dintr-un vârf anume ?

Observații

- Se poate modifica parcurgerea în adâncime, pentru a începe explorarea pe rând, din fiecare vârf
- Pentru graful anterior, se obține rezultatul:
  - AC
  - BACE
  - C
  - DEC
  - EC

Observații

- Prima literă indică vârful de pornire, iar literele următoare arată vârfurile la care se ajunge (fie direct, fie trecând prin alte vârfuri), pornind din vârful de start

Algoritmul lui Warshall

- Algoritmul află dacă **se poate ajunge într-un vârf**, pornind din **oricare alt vârf**
- Se creează un tabel care va indica dacă se poate ajunge într-un vârf, pornind din oricare alt vârf
- Acest tabel se obține prin modificarea matricei de adiacență a grafului

Algoritmul lui Warshall

- Graful reprezentat de această matrice de adiacență revizuită reprezintă **închiderea tranzitivă** a grafului inițial
- În matricea de adiacență, pentru o anumită muchie, linia indică vârful de început al muchiei, iar coloana vârful de sfârșit

	A	B	C	D	E
A	0	0	1	0	0
B	1	0	0	0	1
C	0	0	0	0	0
D	0	0	0	0	1
E	0	0	1	0	0

Observații

- Se poate folosi algoritmul lui Warshall pentru a transforma matricea de adiacență în închiderea tranzitivă a grafului
- Algoritmul se bazează pe următoarea idee: Dacă se poate ajunge de la vârful **L** la vârful **M**, precum și de la vârful **M** la vârful **N**, atunci se poate ajunge de la vârful **L** la vârful **N**

Observații

- Matricea de adiacență arată toate căile posibile cu un singur pas, deci poate fi folosită pentru a obține căi cu doi pași
- Algoritmul construiește căi de lungime arbitrară, bazate pe căi cu mai multe muchii, descoperite anterior

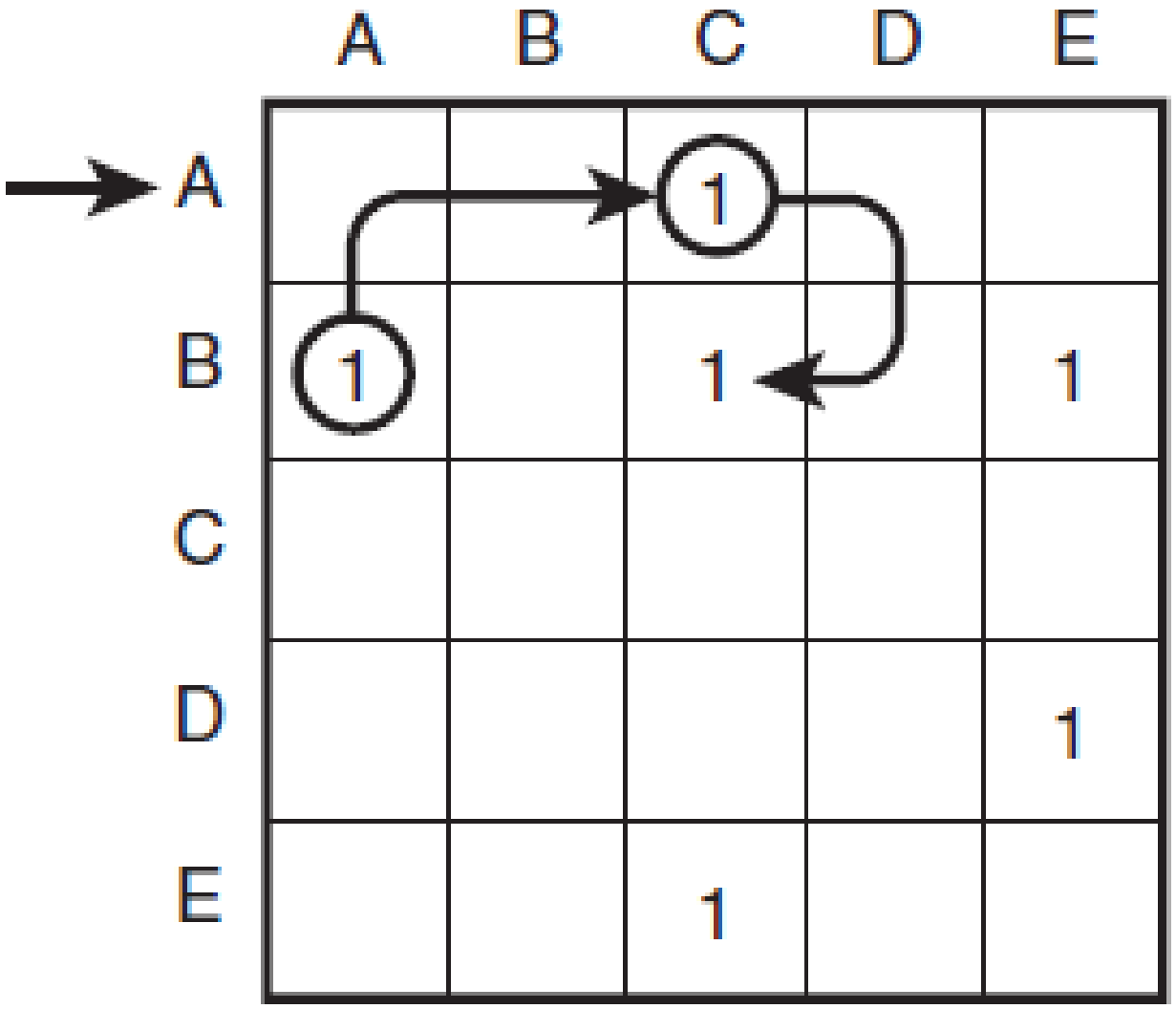
Linia A

- Există 1 pe coloana C, care arată că există o cale de la A la C
- Muchiile posibile care se termină în A se află în coloana A
- Se examinează toate elementele din coloana A
- Se observă că există o muchie de la B la A

Observații

- Există o muchie de la B la A și altă muchie de la A la C
- Se poate deduce că se poate ajunge de la B la C în doi pași
- Se pune 1 la intersecția liniei B cu coloana C

a) y = 0



A to C and B to A
so B to C

Linia B

- Prima celulă din coloana A indică existența unei muchii de la B la A
- Există muchii care se termină în B ?
- Deoarece coloana B conține doar 0-uri, se observă că nicio valoare 1 din linia B nu va duce la obținerea unei căi mai lungi, deoarece nicio muchie nu se termină în B

Liniile C și D

- Linia C nu conține nicio valoare 1
- Se ajunge la linia D, unde se află o muchie de la D la E
- Deoarece coloana D conține numai 0-uri, nu există muchii care se termină în D

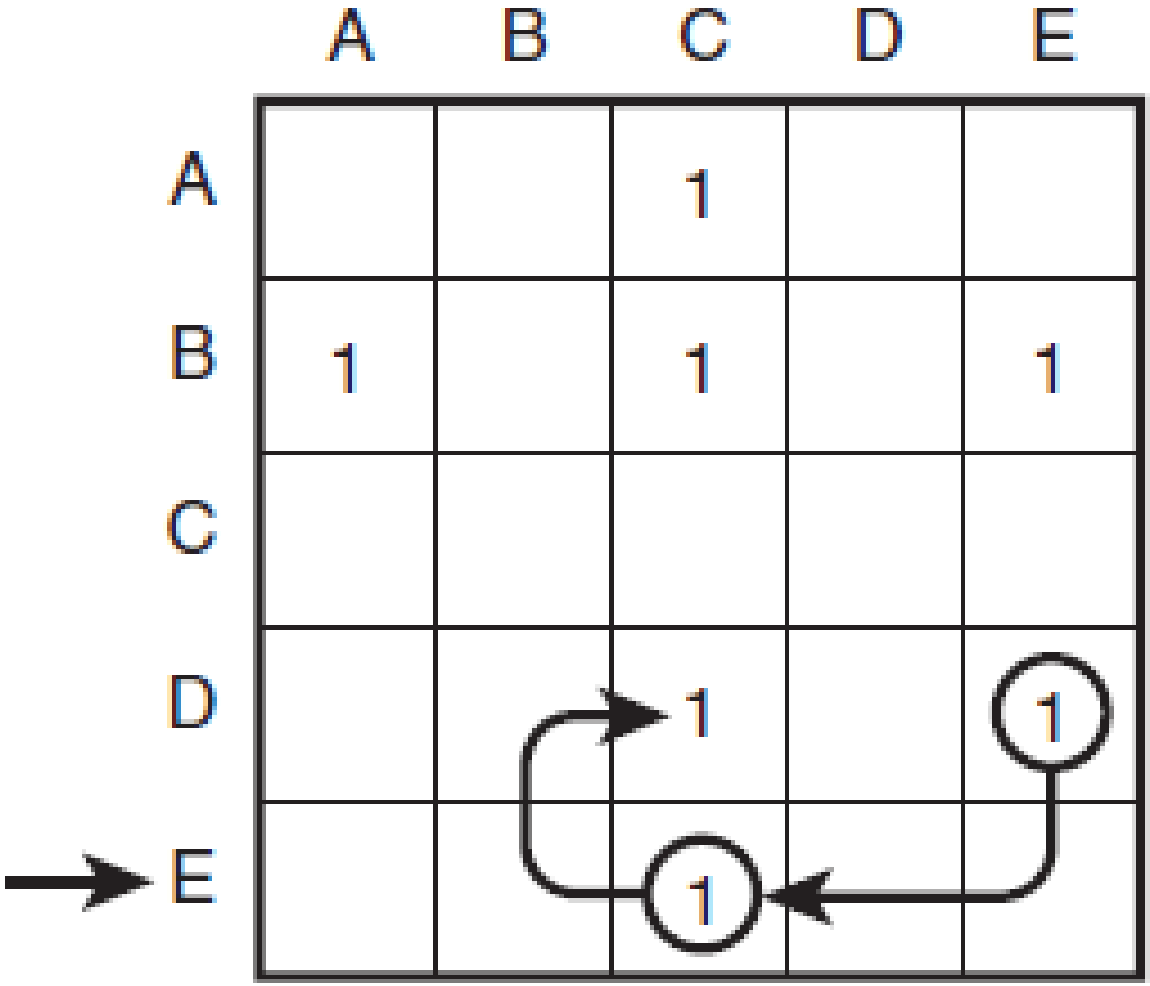
Linia E

- În linia E există o muchie de la E la C
- Din coloana E se observă că există o muchie de la B la E
- Se poate deduce că există o cale de la B la C, deoarece există muchii de la B la E și de la E la C
- Această cale a fost deja descoperită anterior

Observații

- Există 1 în coloana E, corespunzător liniei D
- Muchia de la D la E și de la E la C formează calea de la D la C
- Se adaugă două valori 1 la matricea de adiacență, care arată nodurile la care se poate ajunge dintr-un alt nod, într-un anumit număr de pași

b) y = 4



E to C and D to E
so D to C

Idei de implementare

- Algoritmul folosește trei bucle imbricate
- Bucla exterioară parcurge fiecare linie
- Numim **y** variabila asociată
- Bucla interioară parcurge fiecare celulă de pe linie, folosind variabila **x**
- Dacă se găsește **1** în celula (**y**,**x**), atunci există o muchie de la **y** la **x**

Idei de implementare

- Se activează a treia buclă, cea mai interioară, care are asociată variabila **z**
- A treia buclă examinează celulele din coloana **y**, căutând o muchie care se termină în **y**
- **y** este folosit pentru linii în prima buclă și pentru coloane în a treia buclă

Idei de implementare

- Dacă există **1** la intersecția dintre coloana **y** și linia **z**, atunci există o muchie de la **z** la **y**
- Cu o muchie de la **z** la **y** și altă muchie de la **y** la **x**, se obține calea de la **z** la **x**
- Se pune **1** în celula (**x**,**z**)