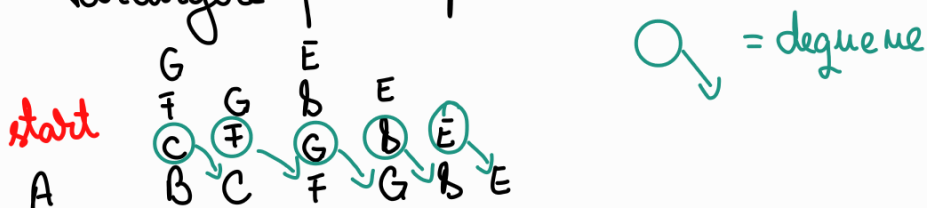


Matrice adiacență

	A	B	C	D	E	F	G
A	0	1	1	0	0	1	1
B	1	0	0	0	0	0	0
C	1	0	0	0	0	0	0
D	0	0	0	0	0	1	0
E	0	0	0	0	0	1	1
F	1	0	0	1	1	0	0
G	1	0	0	0	1	0	0

Parcurgere prin cuprindere:



start : A
dest : b

parent:

	A	B	C	D	E	F	G
	-1	A	A	F	F	A	A

= degenerate

plecăm de la final
A F B
afisare path

Parcurgere în adâncime:

- punem în stivă toți vecinii nodului de start
- vom apela pop pe această stivă și în stivă vom băga toți vecinii acestui nod
- aplicăm recursiv
- când un nod nu are vecini, îi dăm pop la ce avem

1.

B
C
F
G

nod de plecare

A

punem vecinii lui A în stivă
(noduri adiacente)

(le-am pus cât să iasă crescător)

II. $\text{pop}(\text{stivă}) \rightarrow B$

C
F
G

A B

o B nu are vecini

! dacă avea, îi puneam în stivă și aplicam alg.

III. $\text{pop}(\text{stivă}) \rightarrow C$

F
G

A B C

! la fel ca mai sus

IV. $\text{pop}(\text{stivă})$

b
E
G

A B C F

o F are vecini

o îi adăugăm în stivă

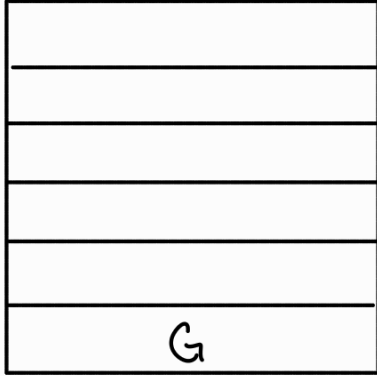
V. $\text{pop} \rightarrow b$

E
G

A B C F b

VI. $\text{pop} \rightarrow E$

A B C F B E

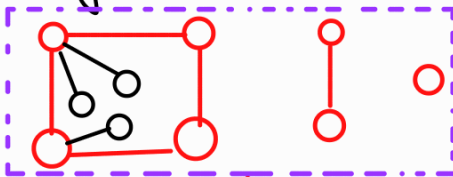


VII. $\text{pop} \rightarrow G$

A B C F B E G



Componente conexe
subgraf cu toate nodurile conectate între ele



\Rightarrow 3 componente conexe