

# ALGORITHMES DE PARCOURS EN PROFONDEUR



# Le Parcours en profondeur

- On explore prioritairement les derniers sommets insérés dans Atteint.
- Tout se passe comme si un seul homme devait visiter l'intégralité du graphe en minimisant son nombre de déplacements.



# Le Parcours en profondeur

- Deux Algorithmes proposés ici :
  - Un itératif, gère Atteint comme une pile.
  - L'autre est un algorithme récursif
- Ce parcours permet :
  - Créer une arborescence
  - Identifier les composantes fortement connexes
  - De sortir d'un labyrinthe planaire (main droite).



# Algorithme de Base (Entête)

- Algorithme VisitGraphProf //Version itérative
  - Données :
    - $G = (X,U)$  un graphe
    - $x$  un sommet de  $G$
  - Donnée/Résultat
    - Exploré : ensemble de sommets
  - Variables
    - $u$  : Sommets de  $G$ ; Atteint : Pile de sommets



# Algorithme de Base (Code)

DébutCode \\ Version itérative

Atteint  $\leftarrow$  PileVide()

Si non( $x \in$  Exploré) alors Empiler( $x$ ,Atteint); Finsi

Tant que Atteint  $\neq$  PileVide() faire

$u \leftarrow$  Sommet(Atteint); Dépiler(Atteint);

    Si non( $u \in$  Exploré) alors

        Exploré  $\leftarrow$  Exploré  $\cup$  { $u$ }; TraiterSommet ( $u$ );

        Pour chaque  $v \in$  Succ( $u$ ) faire

            TraiterArc( $uv$ )

            Si non ( $v \in$  Exploré) alors Empiler( $v$ ,Atteint) finsi

        FinPour

    FinSi

FinTQ

FinCode

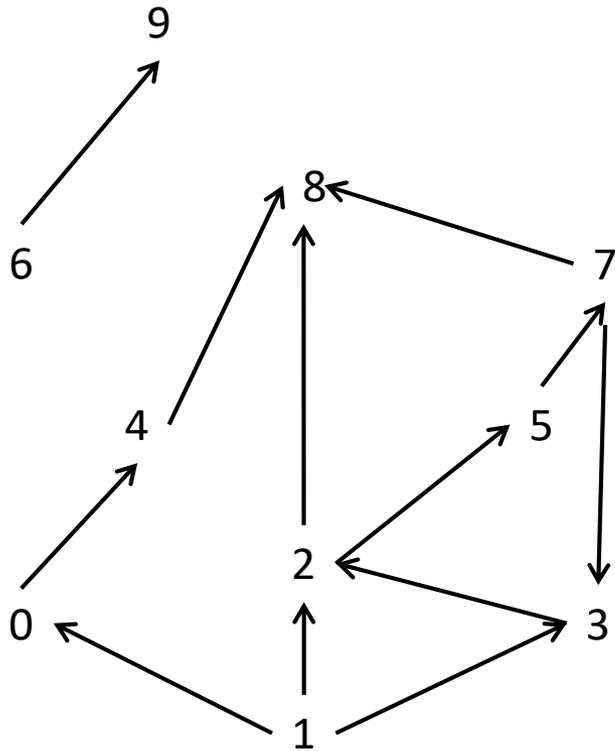


- L'algorithme ParcoursGraphe reste identique pour cette version du parcours en profondeur. Il suffit de changer VisiteGraphe par VisitGraphProf
- L'exécution de l'algorithme sur le graphe exemple est partiellement laissée à titre d'exercice.



# Exemple

## Le graphe



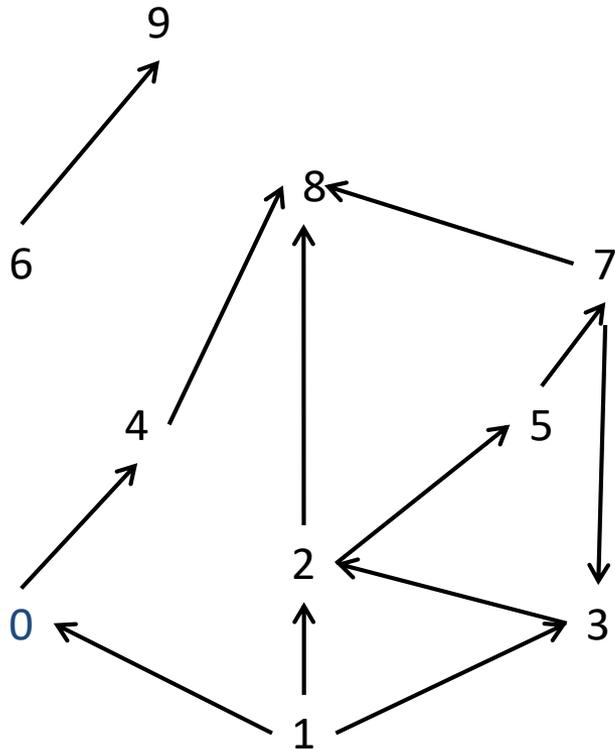
## Les explications

- On exécute `ParcoursGraphes`
- `Exploré = {}`
- `x = 0`
- Appel de `VisitGraphProf` à partir du sommet `x = 0`



# Exemple

## Le graphe



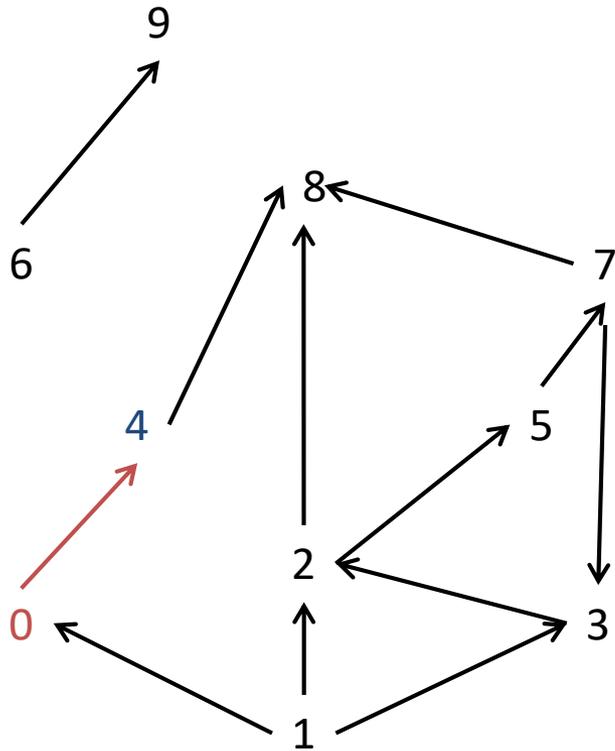
## Les explications

- Exploré = {}
- Atteint = <0>
- On choisi 0, Dépiler 0
- Traitement de 0
- Traitement de l'arc (0,4)
- Empiler 4 Atteint



# Exemple

## Le graphe



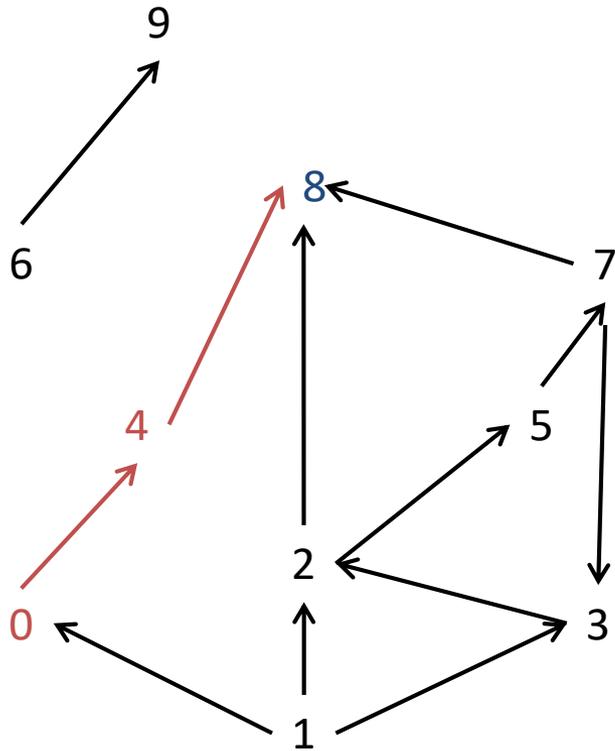
## Les explications

- Exploré = {0}
- Atteint = <4>
- On choisi 4, Dépiler 4
- Traitement de 4
- Traitement de l'arc (4,8)
- Empiler 8



# Exemple

## Le graphe

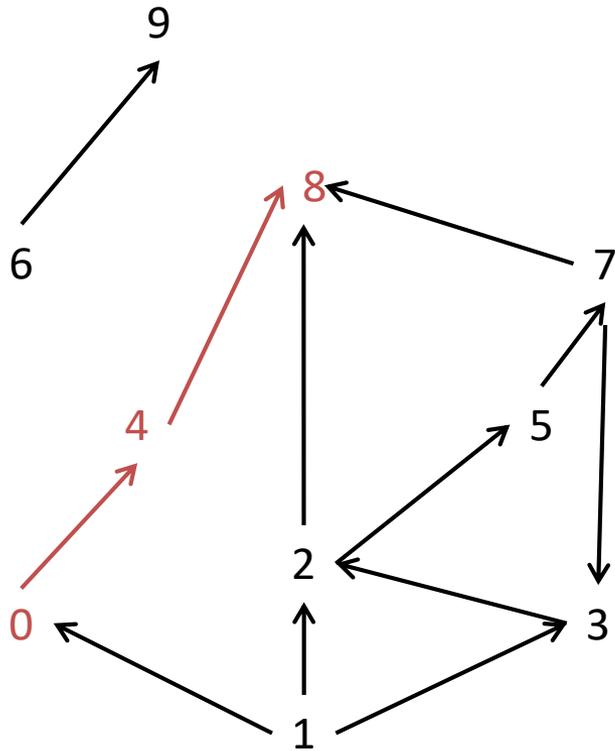


## Les explications

- Exploré = {0, 4}
- Atteint = <8>
- On choisi 8; Dépiler 8
- Traitement de 8
- Fin de VisitGraphProf

# Exemple

## Le graphe

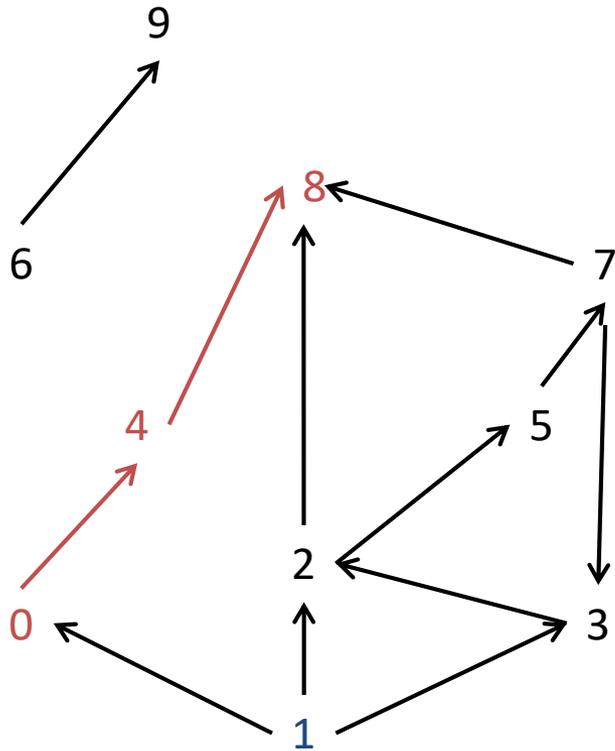


## Les explications

- Exploré = {0, 4, 8}
- Appel de VisitGraphProf avec  $x = 1$

# Exemple

## Le graphe

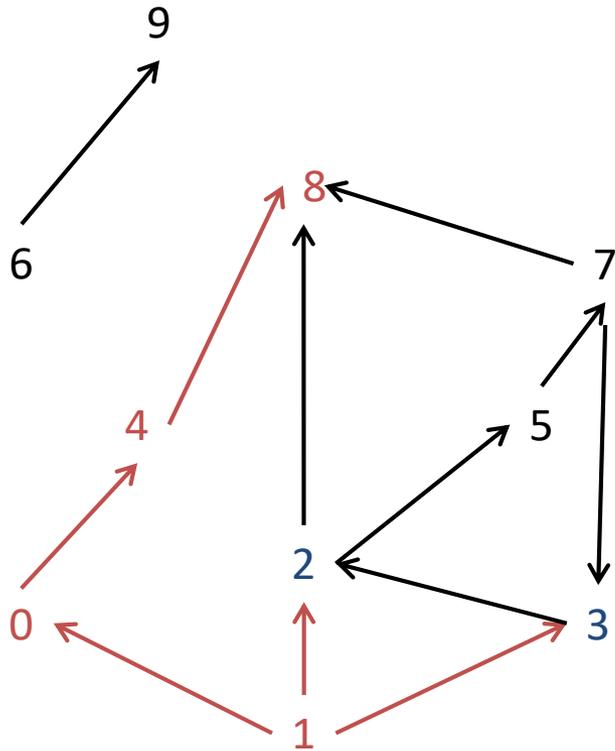


## Les explications

- Exploré = {0, 4, 8}
- Atteint = <1>
- On choisi 1; Dépiler 1
- On traite les arcs (1,0), (1,3), (1,2)
- On empile 3 puis 2 dans Atteint

# Exemple

## Le graphe

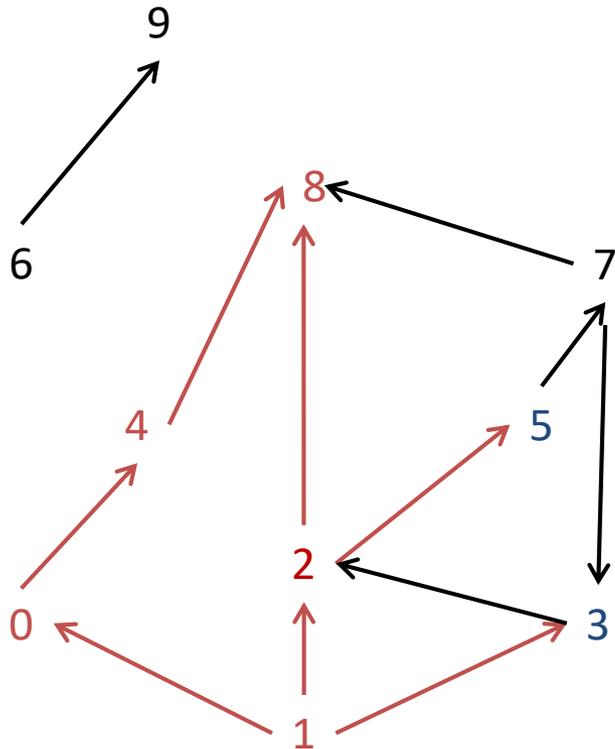


## Les explications

- Exploré = {0, 1, 4, 8}
- Atteint = <2, 3>
- On choisi 2; Dépiler 2
- On traite les arcs (2,5) (2,8)
- On empile 5

# Exemple

## Le graphe

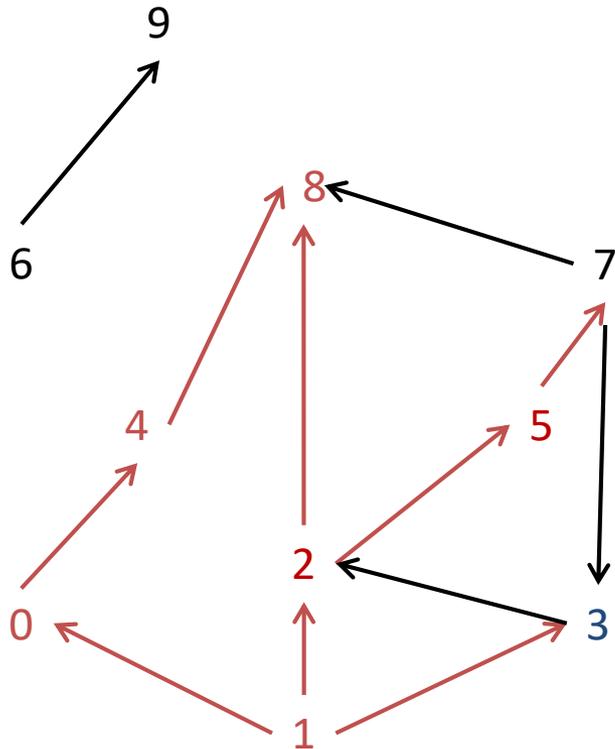


## Les explications

- Exploré = {0, 1, 2, 4, 8}
- Atteint = <5, 3>
- On choisi 5; Dépiler 5
- On traite l'arc (5,7)
- On empile 7

# Exemple

## Le graphe

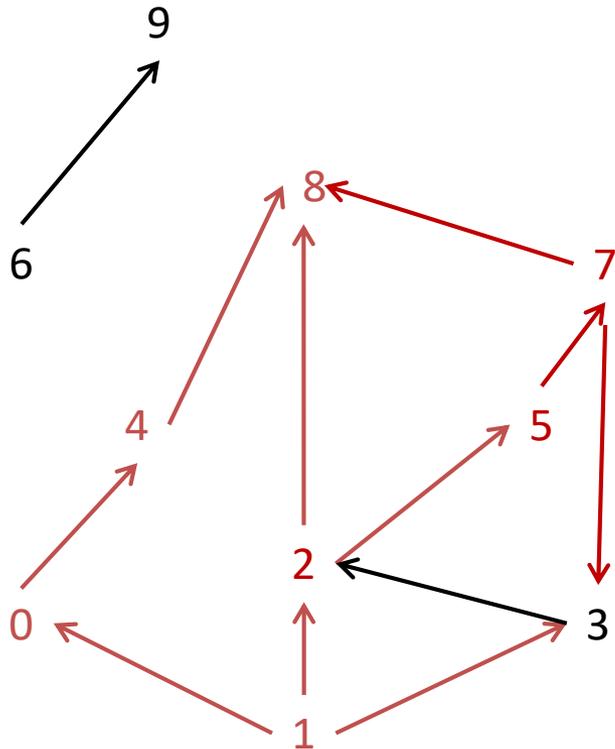


## Les explications

- Exploré = {0, 1, 4, 5, 8}
- Atteint = <7, 3>
- On choisi 7; Dépiler 7
- On traite les arcs (7,3) (7,8)
- On empile 3

# Exemple

## Le graphe

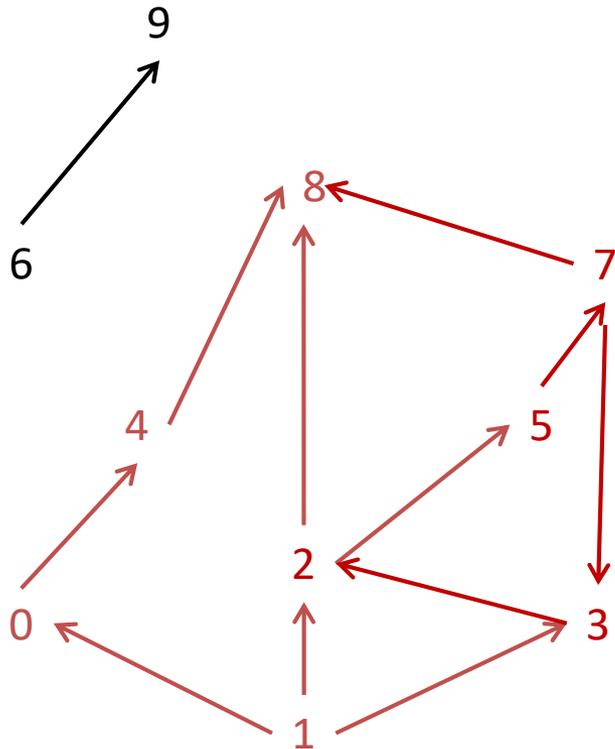


## Les explications

- Exploré = {0,1,2,4,5,7,8}
- Atteint = <3, 3>
- On choisi 3; Dépiler 3
- On traite l'arc (3,2)

# Exemple

## Le graphe



## Les explications

- Exploré = {0,1,2,3,4,5,7,8}
- Atteint = <3>
- On choisi 3; Dépiler 3
- 3 est déjà dans exploré
- Fin de VisitGraphProf
- Je vous laisse finir

# Deuxième Algorithme

- Algorithme récursif
- Donnera en plus deux résultats
  - L'ordre d'insertion des sommets dans la pile des appels récursifs
  - L'ordre de retrait des sommets de la pile des appels récursifs
- Nous utiliserons ces numérotations ultérieurement

# Algorithme de Base (Entête)

- Algorithme VisitGraphProf //Version récursive
  - Données :
    - $G = (X, U)$  un graphe
    - $x$  un sommet de  $G$
  - Donnée/Résultat
    - Exploré : ensemble de sommets
    - $N^{\circ}Emp, N^{\circ}Dep$  : entiers
    - $TEmp, TDep$  : tableaux d'entiers indicés par  $X$
  - Variables
    - $u$  : Sommets de  $G$

# Algorithme de Base (Code)

- DébutCode // Version récursive
  - Si non( $x \in \text{Exploré}$ ) alors
    - $\text{Exploré} \leftarrow \text{Exploré} \cup \{x\}$
    - $\text{TEmp}[x] \leftarrow \text{N}^\circ\text{Emp}; \text{N}^\circ\text{Emp} \leftarrow \text{N}^\circ\text{Emp} + 1$
    - Pour chaque  $u \in \text{Successeur}(x)$  faire
      - $\text{VisitGraphProf}(G, u, \text{Exploré}, \text{N}^\circ\text{Emp}, \text{N}^\circ\text{Dep}, \text{TEmp}, \text{TDep})$
    - FinPour
    - $\text{TDep}[x] \leftarrow \text{N}^\circ\text{Dep}; \text{N}^\circ\text{Dep} \leftarrow \text{N}^\circ\text{Dep} + 1$
  - FinSi
- FinCode

# Algorithme général (entête)

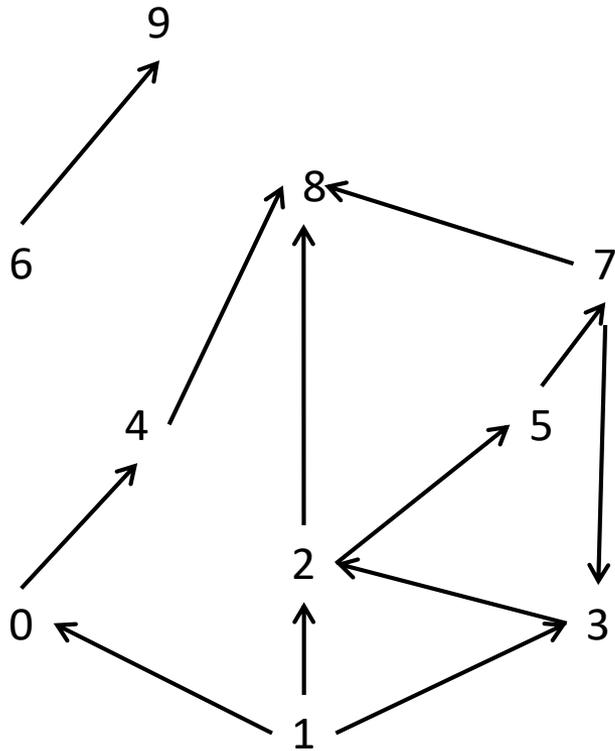
- Algorithme ParcoursGrapheProf
  - Donnée :
    - $G = (X, U)$  : Un graphe
  - Variables :
    - Exploré : ensemble de sommets
    - $x$  : un sommet de  $G$
    - $N^{\circ}\text{Emp}$ ,  $N^{\circ}\text{Dep}$  : entiers
    - $\text{TEmp}$ ,  $\text{TDep}$  : tableaux d'entiers indicés par  $X$

# Algorithme général (Code)

- DébutCode
  - Exploré  $\leftarrow \{\}$ ; N°Emp  $\leftarrow 1$ ; N°Dep  $\leftarrow 1$ ;
  - Pour tout  $x \in X$  faire
    - VisitGraphProf(G,x,Exploré,N°Emp,N°Dep,TEmp,TDep)
  - FinPour
- FinCode

# Exemple

## Le graphe

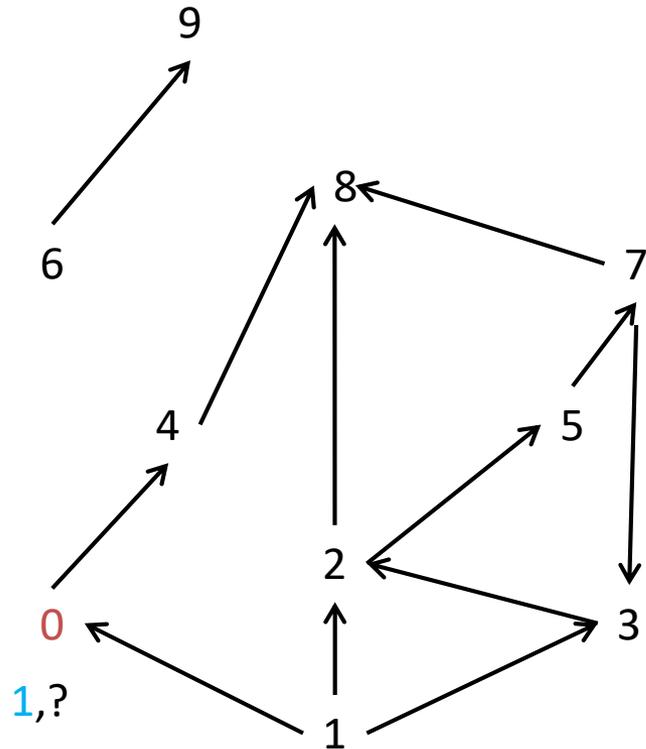


## Les explications

- On exécute `ParcoursGraphesProf`
- `Exploré = {}`
- `x = 0,`
- `N°Emp = 1; N°Dep=1`
- Appel de `VisitGraphProf` à partir du sommet `x = 0`

# Exemple

## Le graphe

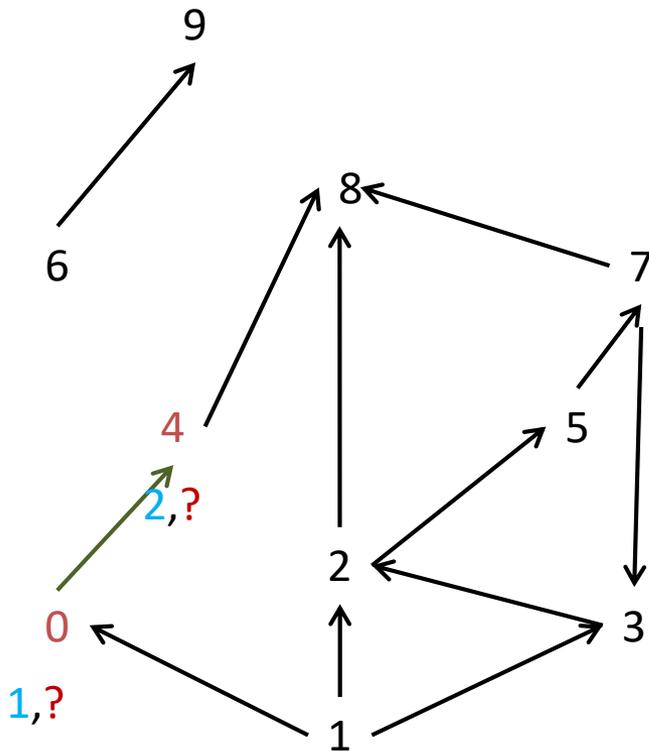


## Les explications

- Exploré = {0}
- N°Emp = 2; N°Dep=1
- Appel de VisitGraphProf à partir du sommet x = 4

# Exemple

## Le graphe

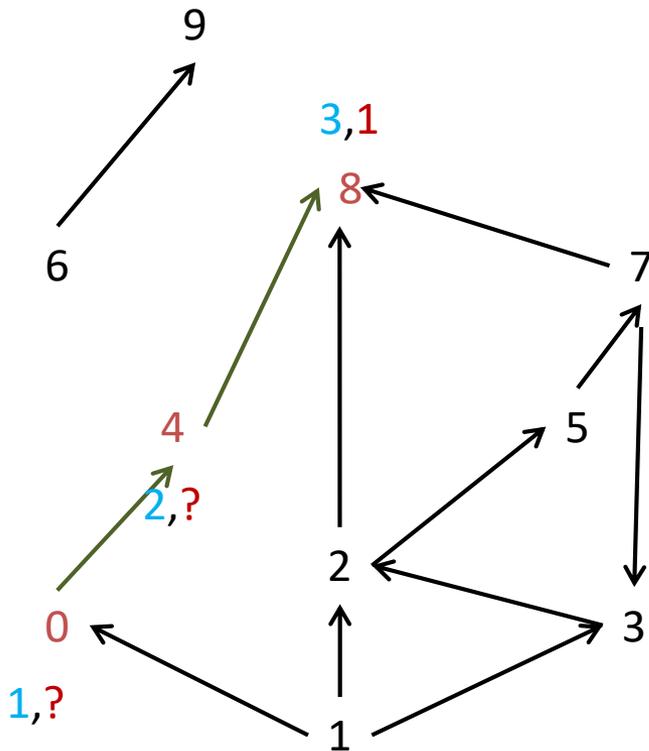


## Les explications

- Exploré = {0,4}
- N°Emp = 3; N°Dep=1
- Appel de VisitGraphProf à partir du sommet x = 8

# Exemple

## Le graphe

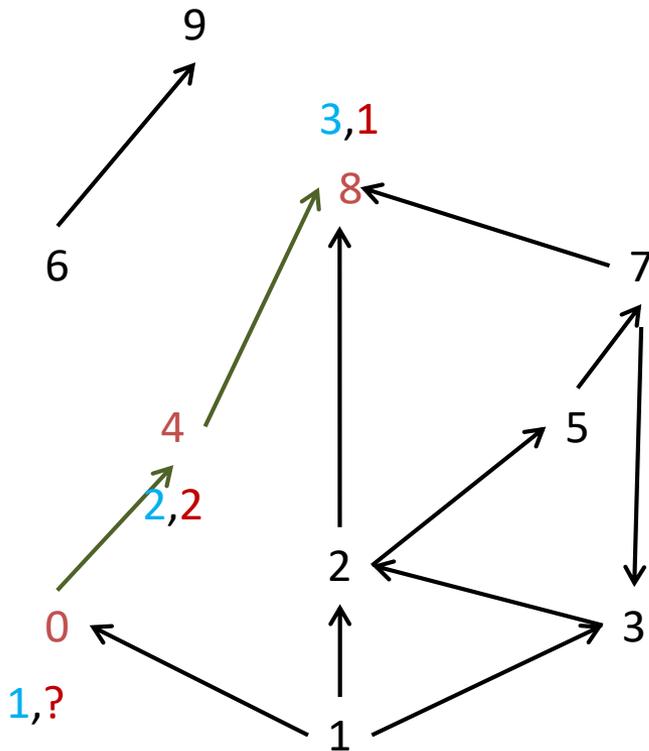


## Les explications

- Exploré = {0,4,8}
- N°Emp = 4 ; N°Dep=2
- Retour vers VisitGraphProf à partir du sommet x = 4

# Exemple

## Le graphe

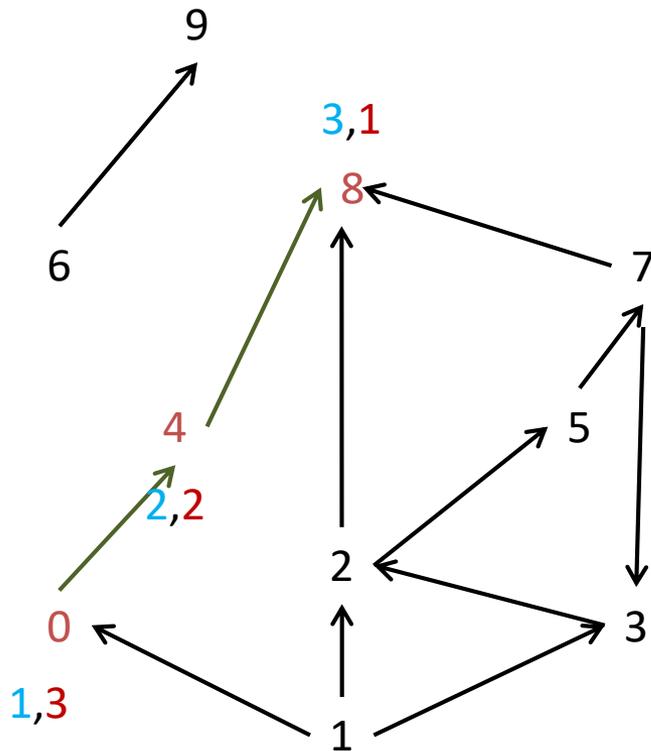


## Les explications

- Exploré = {0,4,8}
- N°Emp = 4 ; N°Dep=3
- Retour vers VisitGraphProf à partir du sommet x = 0

# Exemple

## Le graphe

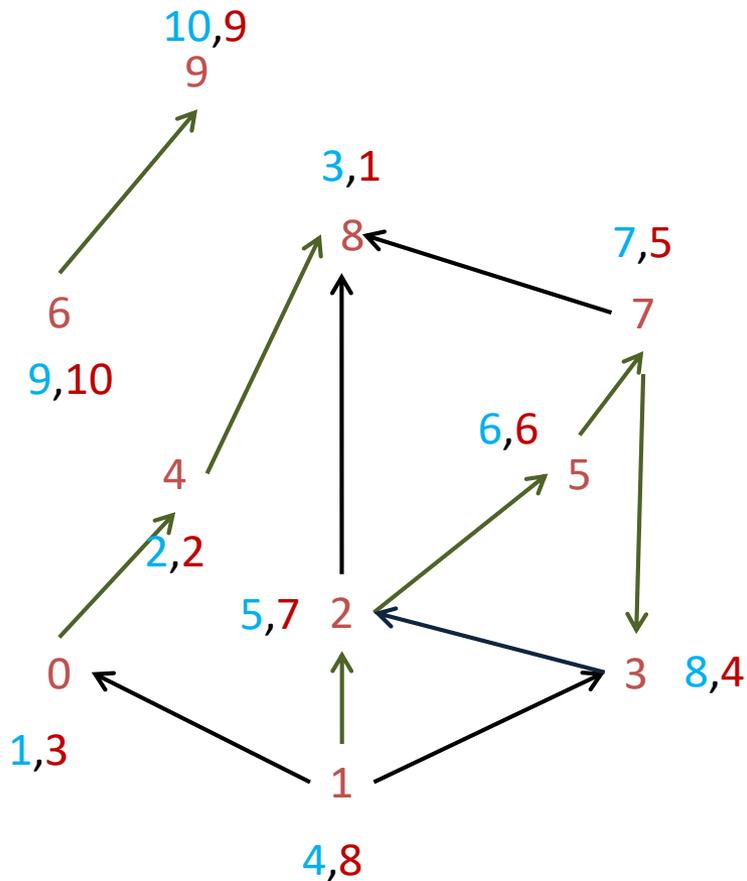


## Les explications

- Exploré = {0,4,8}
- N°Emp = 4 ; N°Dep=4
- Retour vers  
ParcoursGrapheProf qui  
relance la visite à partir de x  
= 1

# Exemple

## Le graphe



## Les explications

- Exploré = {0,4,8}
- N°Emp = 4 ; N°Dep = 4
- Retour vers  
ParcoursGrapheProf qui  
relance la visite à partir de x  
= 1
- Et cetera

# Classons les arcs

- Les arcs  $(x,y)$  dont le numéro d'empilement de l'origine  $x$  est inférieur au numéro d'empilement de l'extrémité  $y$  et dont le numéro de dépilement de l'origine est supérieur au numéro de dépilement de l'extrémité.
- $TEmp[x] < TEmp [y]$  et  $TDep[x] > TDep [y]$
- Sont les arcs de l'arbre ou les arcs de transitivité

# Classons les arcs

- Les arcs  $(x,y)$  dont le numéro d'empilement de l'origine  $x$  est Supérieur au numéro d'empilement de l'extrémité  $y$  et dont le numéro de dépilement de l'origine est supérieur au numéro de dépilement de l'extrémité.
- $TEmp[x] > TEmp [y]$  et  $TDep[x] > TDep [y]$
- Sont les arcs traversiers

# Classons les arcs

- Les arcs  $(x,y)$  dont le numéro d'empilement de l'origine  $x$  est Supérieur au numéro d'empilement de l'extrémité  $y$  et dont le numéro de dépilement de l'origine est inférieur au numéro de dépilement de l'extrémité.
- $TEmp[x] > TEmp [y]$  et  $TDep[x] < TDep [y]$
- Sont les arcs qui ferment un circuit

# Conséquence

- Cette classification des arcs permet de calculer facilement les composantes fortement connexes du graphe.
- Sur les graphes non orienté cet algorithme permet de calculer avec ces mêmes numérotations les isthmes et les poids d'articulations et même de reconnaître les graphes planaires.