

Composantes fortement connexes

Algorithmique – L3

François Laroussinie

15 novembre 2010

Plan

- 1 Définitions
- 2 La définition des coefficients
- 3 Calcul des coefficients

Plan

- 1 Définitions
- 2 La définition des coefficients
- 3 Calcul des coefficients

Composantes fortement connexes

$G = (S, A)$: un graphe **orienté**.

Définition : Une composante fortement connexe (CFC) \mathcal{C} de G est un **sous-ensemble maximal** de sommets de G tel que :
si $u, v \in \mathcal{C}$, alors $u \rightarrow_G^* v$ et $v \rightarrow_G^* u$.

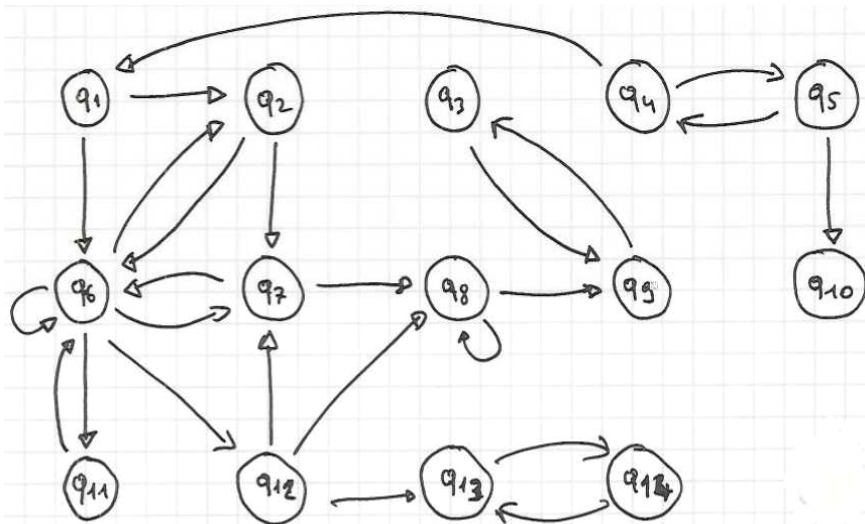
Composantes fortement connexes

$G = (S, A)$: un graphe **orienté**.

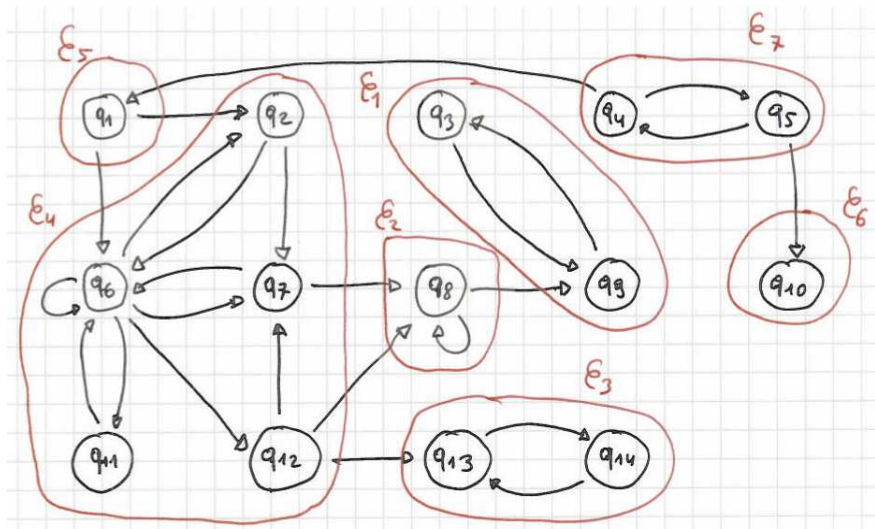
Définition : Une composante fortement connexe (CFC) \mathcal{C} de G est un **sous-ensemble maximal** de sommets de G tel que :
si $u, v \in \mathcal{C}$, alors $u \rightarrow_G^* v$ et $v \rightarrow_G^* u$.

Algorithme de Tarjan (SIAM Journal of Computing, Vol. 1, No. 2, June 1972)

Exemple : le graphe G



Exemple : les CFC de G



Idée de l'algorithme

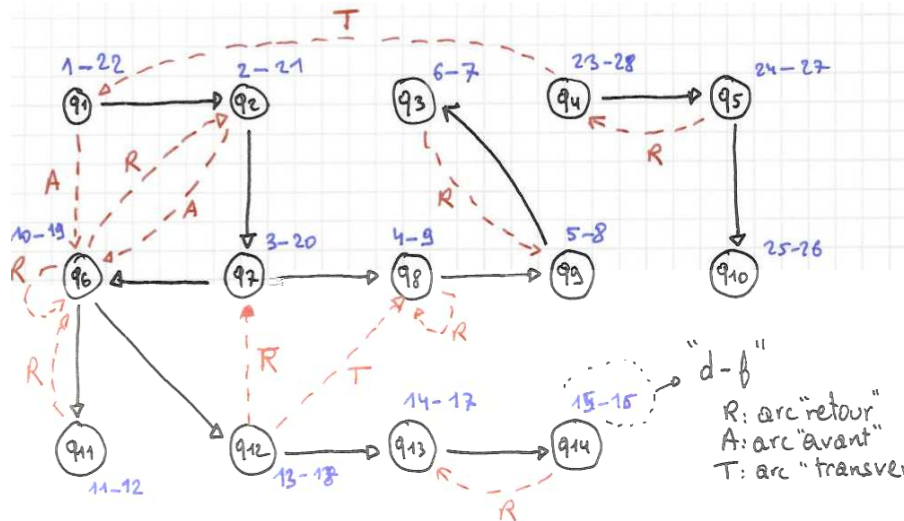
L'algorithme de recherche des CFC est basé sur les dates $d[-]$ et la forêt G_{Π} calculées lors d'un parcours en profondeur.

Idée de l'algorithme

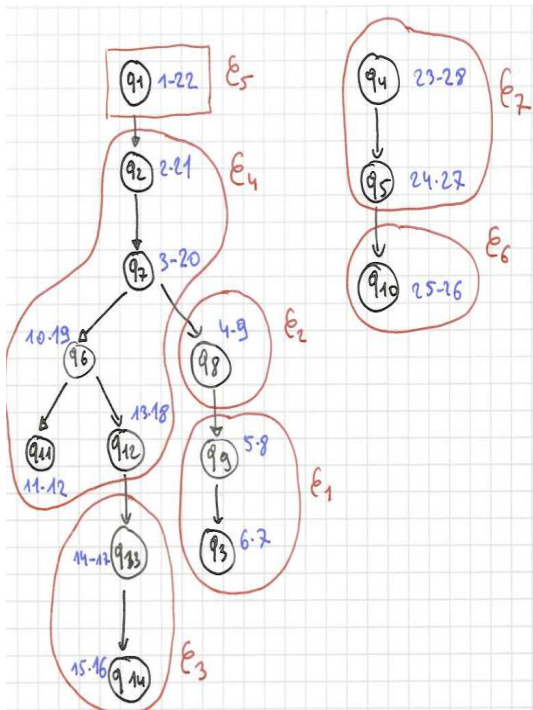
L'algorithme de recherche des CFC est basé sur les dates $d[-]$ et la forêt G_{Π} calculées lors d'un parcours en profondeur.

exemple. . .

Exemple : un parcours en profondeur de G



On en déduit la forêt $G_{\Pi} \dots$



Observations

Les sommets d'une même CFC n'apparaissent pas n'importe comment dans la forêt G_{Π} ...

Est-ce un hasard ?

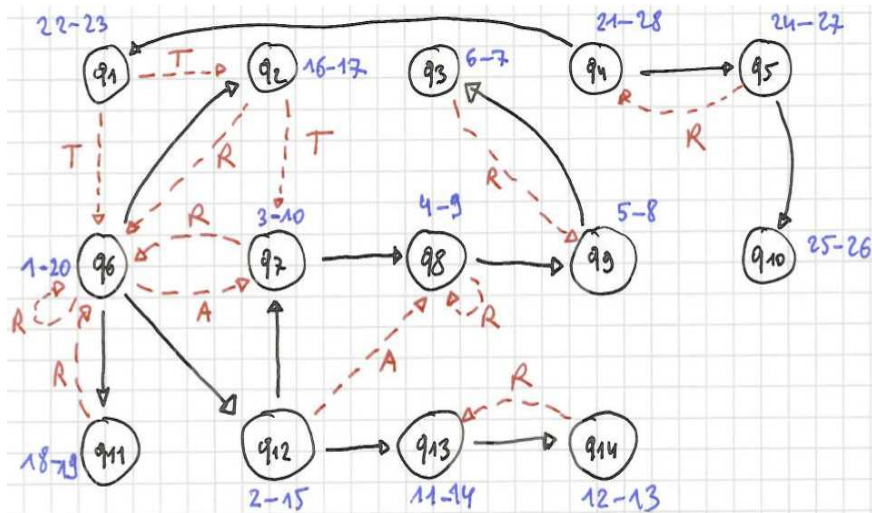
Observations

Les sommets d'une même CFC n'apparaissent pas n'importe comment dans la forêt G_{Π} ...

Est-ce un hasard ?

Autre essai...

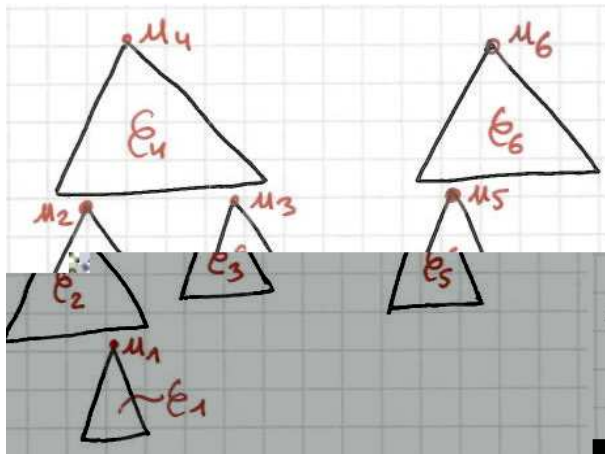
Autre parcours



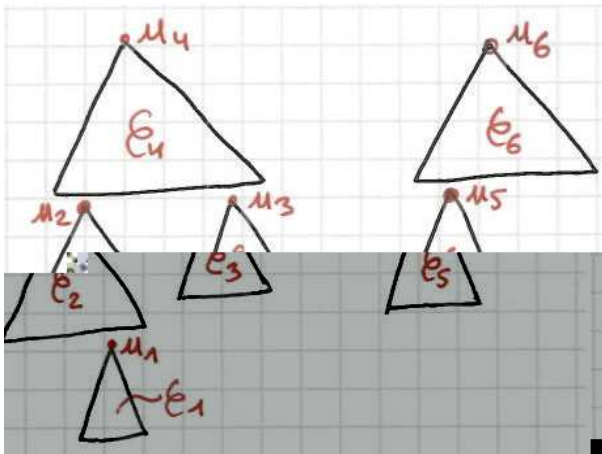
Propriété 21 Soient $v, w \in S$ tels que $v \leftrightarrow_G^* w$,

- ① alors v et w ont un ancêtre commun dans G_Π ; et
- ② soit u le sommet tel que (1) $u \rightarrow_{G_\Pi}^* v$, (2) $u \rightarrow_{G_\Pi}^* w$ et (3) $d[u]$ est maximal, alors on a : $v \leftrightarrow_G^* u$ et $u \leftrightarrow_G^* w$.

Place des CFC dans la forêt G_{Π}



Place des CFC dans la forêt G_{Π}



Corollaire :

Soit \mathcal{C} une CFC de G .

Alors l'arborescence G_{Π} restreinte aux sommets de \mathcal{C} est un arbre (couvrant \mathcal{C}).

Idée générale de l'algorithme

\Rightarrow calculer un **coefficient** $r[x]$ pour chaque $x \in S$ à partir des dates $d[-]$ de manière à ce que $\forall x \in S$, on ait :
 $r[x] = d[x]$ ssi x est la racine de sa CFC

Def : la racine d'une CFC \mathcal{C} est le premier sommet de \mathcal{C} à être découvert dans le parcours en profondeur.

Idée générale de l'algorithme

\Rightarrow calculer un coefficient $r[x]$ pour chaque $x \in S$ à partir des dates $d[-]$ de manière à ce que $\forall x \in S$, on ait :
 $r[x] = d[x]$ ssi x est la racine de sa CFC

Def : la racine d'une CFC \mathcal{C} est le premier sommet de \mathcal{C} à être découvert dans le parcours en profondeur.

Les coefficients $r[-]$ vous nous permettent de trouver les racines des CFC, puis de trouver les autres sommets de ces CFC.

Plan

- 1 Définitions
- 2 La définition des coefficients
- 3 Calcul des coefficients

Définition des coefficients r

$\forall x \in S$, on définit le coefficient $r[x]$ par :

$$r[x] \stackrel{\text{def}}{=} \min \left(\{d[x]\} \cup \{d[w] \mid \begin{array}{l} (1) \ x \rightarrow_{G_\Pi}^* w \xrightarrow{T/R} w \wedge \\ (2) \ \exists u \in S. u \rightarrow_{G_\Pi}^* x \wedge u \rightarrow_{G_\Pi}^* w \wedge (3) u \leftrightarrow_G^* w \end{array} \} \right)$$

Définition des coefficients r

$\forall x \in S$, on définit le coefficient $r[x]$ par :

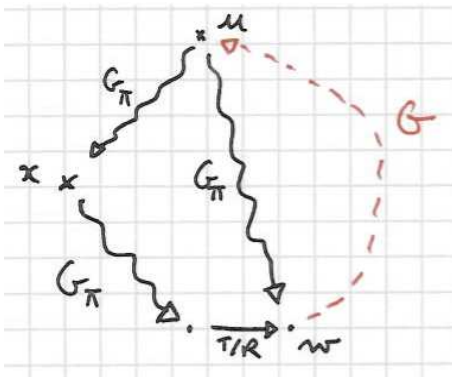
$$r[x] \stackrel{\text{def}}{=} \min \left(\{d[x]\} \cup \{d[w] \mid \begin{array}{l} \text{(1) } x \rightarrow_{G_\Pi}^* w \xrightarrow{T/R} w \wedge \\ \text{(2) } \exists u \in S. u \rightarrow_{G_\Pi}^* x \wedge u \rightarrow_{G_\Pi}^* w \wedge \text{(3) } u \leftrightarrow_G^* w \end{array} \} \right)$$

$r[x]$ est le minimum entre $d[x]$ et les dates de découverte des sommets w qui :

- ① sont accessibles depuis x par un chemin de G_Π prolongé par **un** arc “retour” ou un arc “transverse”, et tels que
- ② x et w ont un ancêtre commun dans G_Π , et
- ③ u et w sont dans la même CFC.

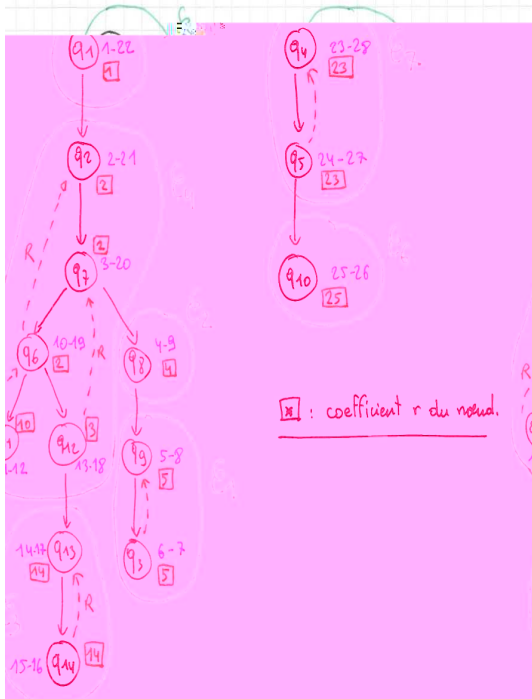
Définition des coefficients

$$r[x] \stackrel{\text{def}}{=} \min \left(\{d[x]\} \cup \{d[w] \mid \textcolor{blue}{(1)} x \rightarrow_{G_\Pi}^* w \xrightarrow{T/R} w \wedge \right. \\ \left. \textcolor{blue}{(2)} \exists u \in S. u \rightarrow_{G_\Pi}^* x \wedge u \rightarrow_{G_\Pi}^* w \wedge \textcolor{blue}{(3)} u \leftrightarrow_G^* w \} \right)$$



Autre définition

$r[x]$ est le minimum entre $d[x]$ et les $d[w]$ tel que w est un sommet de la CFC de x accessible depuis x par un chemin de la forme $x \rightarrow_{G_\Pi}^* \xrightarrow{T/R} w$.



Comme w , u et x sont dans la même CFC, on a :

Propriété 22

Pour tout sommet x d'une CFC \mathcal{C} de racine $u_{\mathcal{C}}$, on a :

$$d[u_{\mathcal{C}}] \leq r[x] \leq d[x]$$

Comme w , u et x sont dans la même CFC, on a :

Propriété 22

Pour tout sommet x d'une CFC \mathcal{C} de racine $u_{\mathcal{C}}$, on a :

$$d[u_{\mathcal{C}}] \leq r[x] \leq d[x]$$

Propriété 23

Un sommet x est la racine d'une composante fortement connexe **si et seulement si** $r(x) = d[x]$.

Plan

1 Définitions



Algorithme de Tarjan

On modifie l'algorithme de parcours en profondeur...

Les nouvelles variables utilisées sont :

- $nbcfc$: pour compter le nombre de CFC trouvées ;
- $NumCFC[-]$: un tableau donnant pour chaque sommet le numéro de sa CFC ;
- $r_A[-]$: un tableau pour stocker les coefficients r ;
- P est une pile pour stocker des sommets de G .

Algorithme de Tarjan

Procédure Tarjan-CFC(G)

$temps := 0$;

$nbcfc := 0$;

$P :=$ Pile vide;

pour chaque $x \in S$ **faire**

 Couleur[x] := blanc;

 NumCFC[x] := undef;

pour chaque $x \in S$ **faire**

si Couleur[x] = blanc **alors** CFC(x);

retourner NumCFC[]

```

1  Procédure CFC( $x$ )
2   $temps++$ ;  $d[x] := temps$ ;  $Couleur[x] := \text{gris}$ ;
3   $r_A[x] := d[x]$ ;
4   $P.\text{Empiler}(x)$ ;
5  pour chaque  $(x, y) \in A$  faire
6      si  $Couleur[y] = \text{blanc}$  alors
7           $CFC(y)$ ;
8           $r_A[x] := \min(r_A[x], r_A[y])$ ;
9      sinon
10         si  $d[y] < d[x] \wedge y \in P$  alors
11              $r_A[x] := \min(r_A[x], d[y])$ ;
12  $temps++$ ;  $f[x] := temps$ ;  $Couleur[x] := \text{noir}$ ;
13 si  $r_A[x] = d[x]$  alors
14      $nbcfc++$ ;
15     tant que  $P \neq \emptyset \wedge d[P.Tete()] \geq d[x]$  faire
16          $y := P.Tete()$ ;
17          $P.Depiler()$ ;
18          $NumCFC[y] := nbcfc$ ;

```


Comment ça marche ?

Il faut montrer que les coefficients $r_A[-]$ calculés par l'algorithme sont égaux aux $r[-]$ définis précédemment.

Quelques éléments. . .

Comment ça marche ?

Il faut montrer que les coefficients $r_A[-]$ calculés par l'algorithme sont égaux aux $r[-]$ définis précédemment.

Quelques éléments. . .

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π .

Les sommets de \mathcal{C} ne sont extraits de P que lorsque $u_{\mathcal{C}}$ est coloriée en noir et que les instructions 14-18 de CFC sont exécutées.

Comment ça marche ?

Il faut montrer que les coefficients $r_A[-]$ calculés par l'algorithme sont égaux aux $r[-]$ définis précédemment.

Quelques éléments. . .

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π .
Les sommets de \mathcal{C} ne sont extraits de P que lorsque $u_{\mathcal{C}}$ est coloriée en noir et que les instructions 14-18 de CFC sont exécutées.
- l'instruction 8 permet de faire "remonter" les valeurs de $r_A[y]$ à son père x dans G_Π .

Comment ça marche ?

Il faut montrer que les coefficients $r_A[-]$ calculés par l'algorithme sont égaux aux $r[-]$ définis précédemment.

Quelques éléments. . .

- P contient les sommets dont la racine de leur CFC est encore grise. Ils ont des ancêtres communs dans G_Π .
Les sommets de \mathcal{C} ne sont extraits de P que lorsque $u_{\mathcal{C}}$ est coloriée en noir et que les instructions 14-18 de CFC sont exécutées.
- l'instruction 8 permet de faire "remonter" les valeurs de $r_A[y]$ à son père x dans G_Π .
- le test de la ligne 10 (" $d[y] < d[x] \wedge y \in P$ ") caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow_G^* y$? ...

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow_G^* y$.

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_{Π} et $x \leftrightarrow_G^* y$.

- $(x, y) : \notin G_{\Pi}$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$);

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_{Π} et $x \leftrightarrow_G^* y$.

- $(x, y) : \notin G_{\Pi}$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$);
- $x, y \in P \Rightarrow$ ils ont des ancêtres communs dans G_{Π} ;

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_{Π} et $x \leftrightarrow_G^* y$.

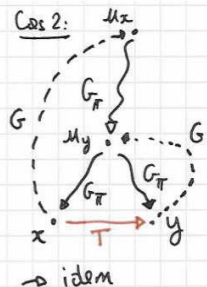
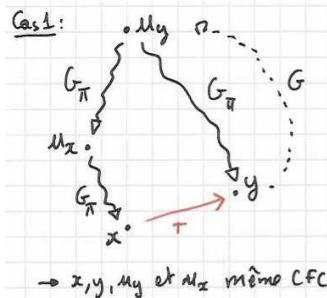
- $(x, y) : \notin G_{\Pi}$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$);
- $x, y \in P \Rightarrow$ ils ont des ancêtres communs dans G_{Π} ;
- $x \leftrightarrow_G^* y$:

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_{Π} et $x \leftrightarrow_G^* y$.

- $(x, y) : \notin G_{\Pi}$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$);
- $x, y \in P \Rightarrow$ ils ont des ancêtres communs dans G_{Π} ;
- $x \leftrightarrow_G^* y$:
 - direct si (x, y) est un arc retour...

le test " $d[y] < d[x] \wedge y \in P$ " caractérise bien les arcs (x, y) de type "retour" ou "transverse" t.q. y a des ancêtres communs avec x dans G_Π et $x \leftrightarrow_G^* y$.

- $(x, y) : \notin G_\Pi$ (y n'est pas blanc) et pas arc avant ($d[y] < d[x]$);
- $x, y \in P \Rightarrow$ ils ont des ancêtres communs dans G_Π ;
- $x \leftrightarrow_G^* y$:
 - direct si (x, y) est un arc retour...
 - si (x, y) est un arc transverse, soient u_x et u_y les racines (en **gris** car dans P) de leurs CFC...



Correction de l'algorithme de Tarjan

Théorème

L'algorithme de Tarjan, vérifie les propriétés suivantes :

- ① à la date $f[x]$, on a : $r_A[x] = r[x]$,
- ② à la fin du traitement $CFC(u_C)$, tous les sommets y de \mathcal{C} sont retirés de la pile P et vérifient : $NumCFC[y] = NumCFC[u_C]$,
- ③ à la fin de $Tarjan-CFC(G)$, chaque CFC a reçu un numéro différent : pour deux CFC distinctes \mathcal{C} et \mathcal{C}' , on a $NumCFC[u_C] \neq NumCFC[u_{C'}]$.