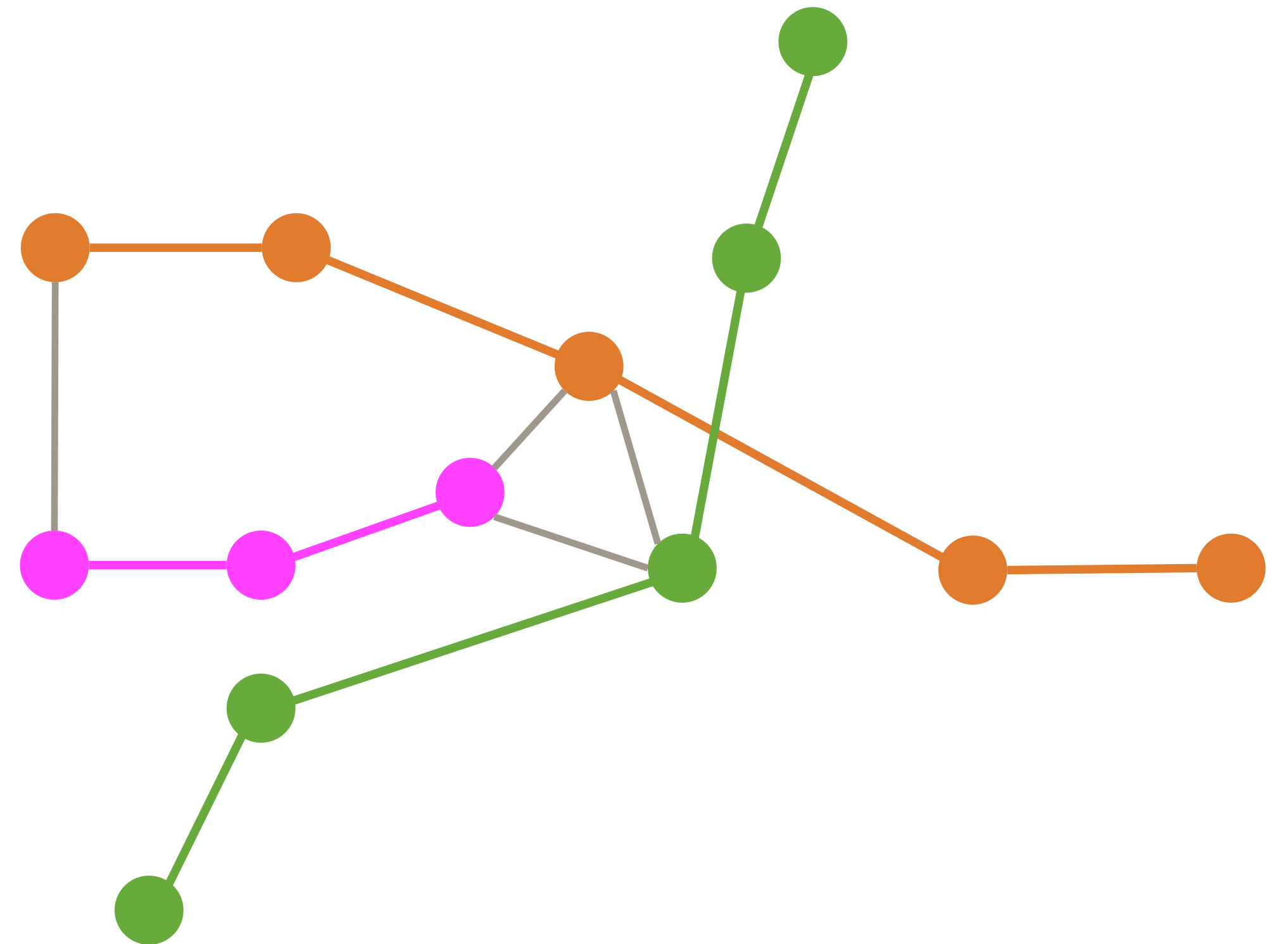
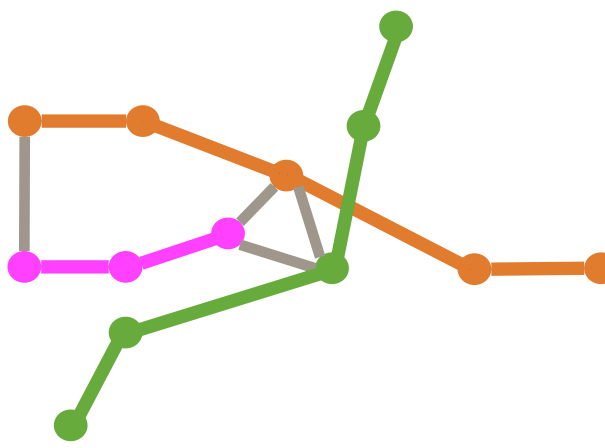


6.8. Composantes fortement connexes d'un graphe orienté

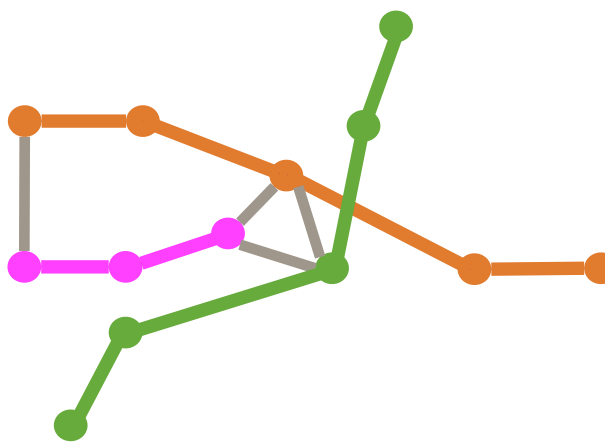


Connectivité forte

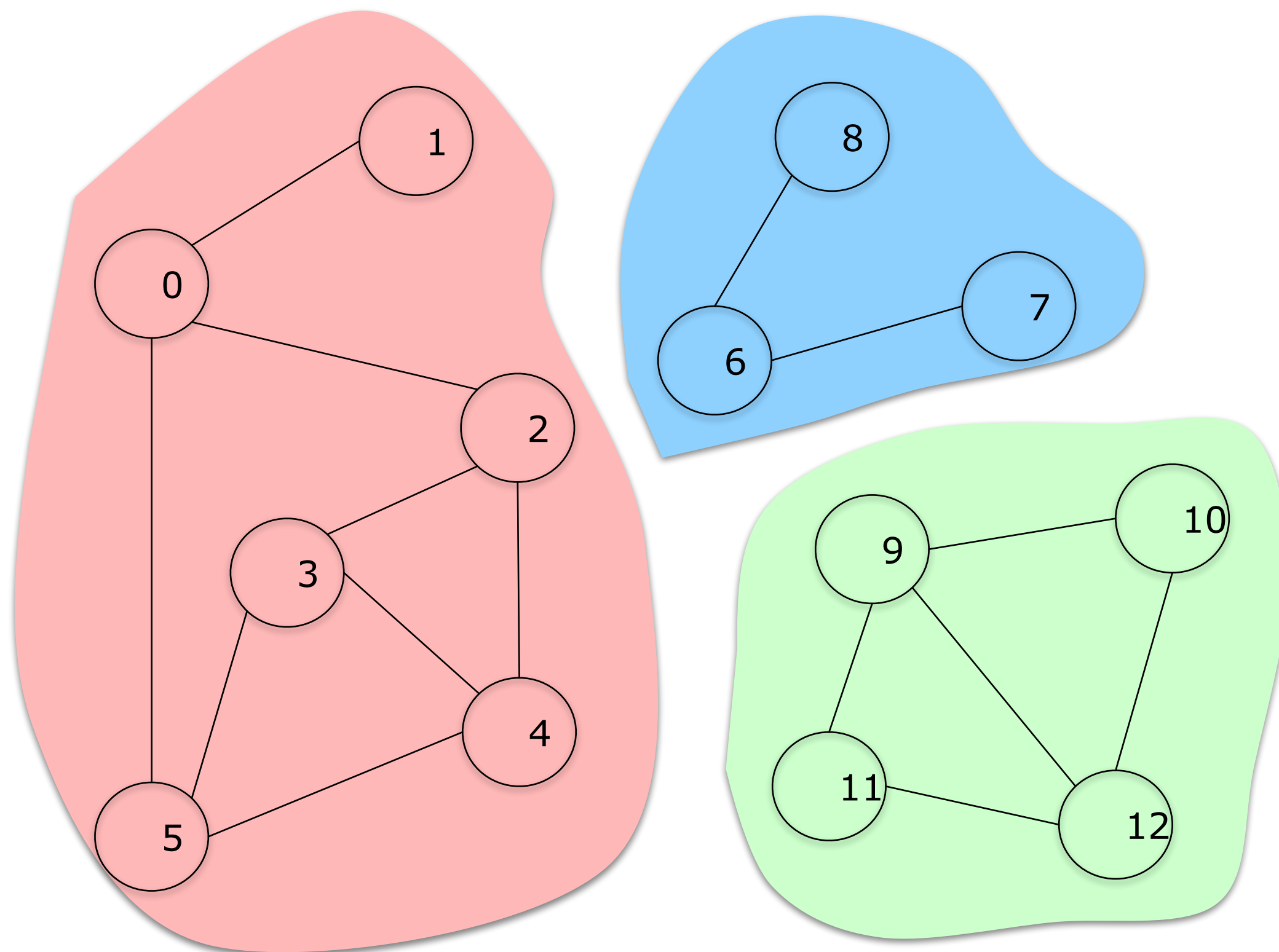


- **Définition** – les sommets v et w sont **fortement connectés** s'il existe un chemin (orienté) de v à w **et** de w à v .
- **Propriété** – la connectivité forte est une relation d'équivalence, i.e. elle est réflexive, symétrique et transitive
- **Définition** – une **composante fortement connexe** (Strongly Connected Component) CFC est un ensemble maximal de sommets fortement connectés.

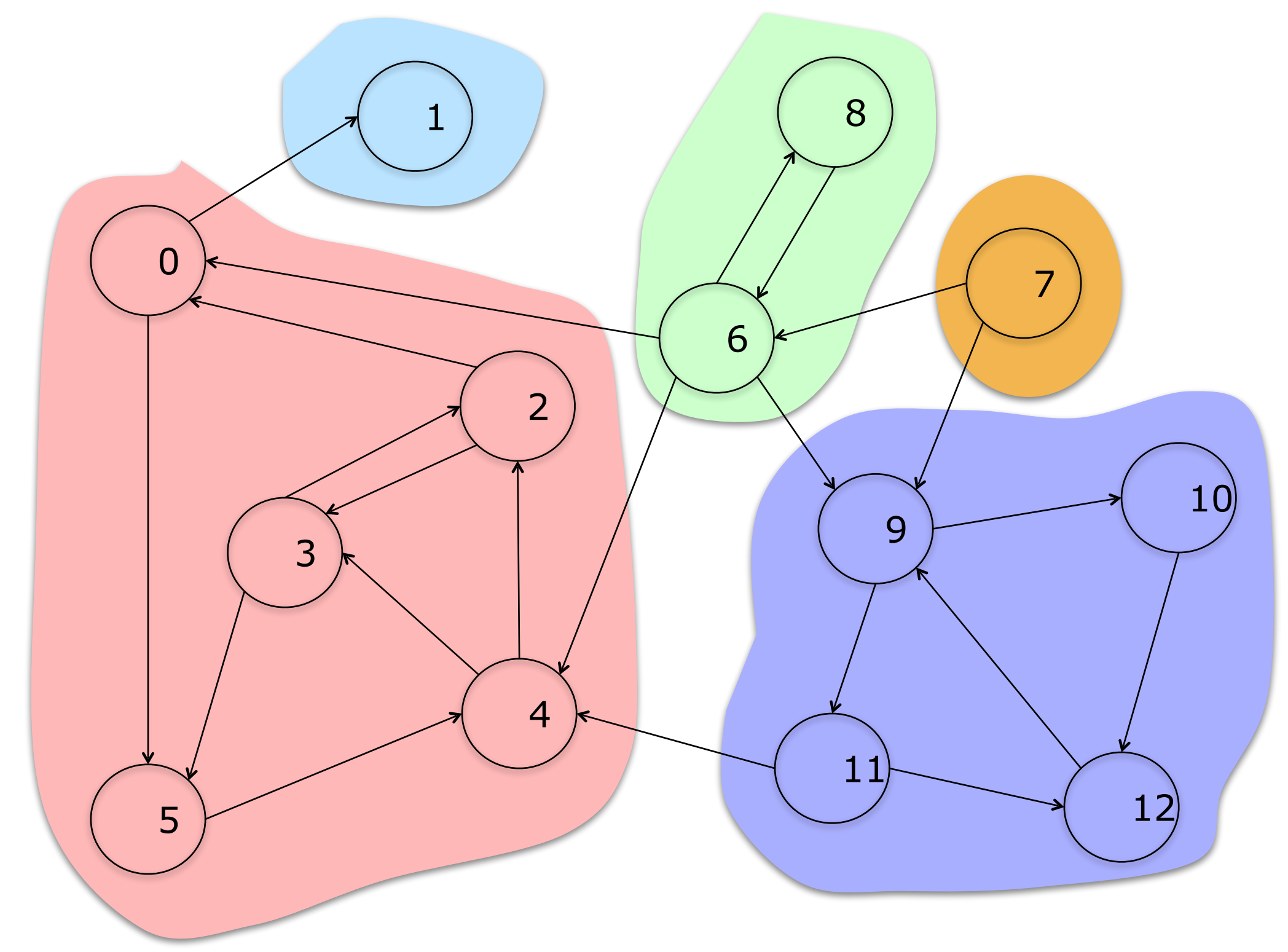
CC vs. CFC



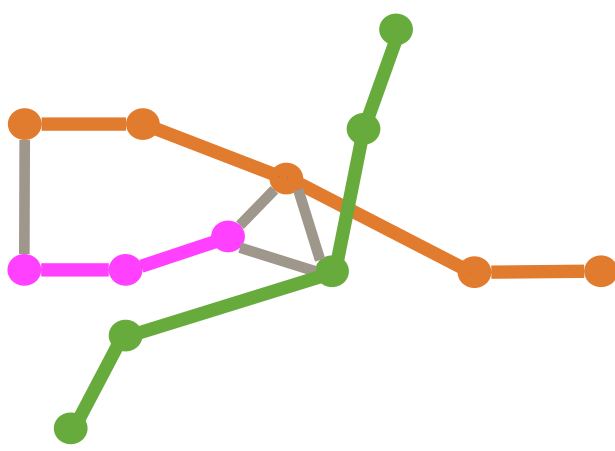
v et w sont connectés s'il y a une chaîne reliant v et w



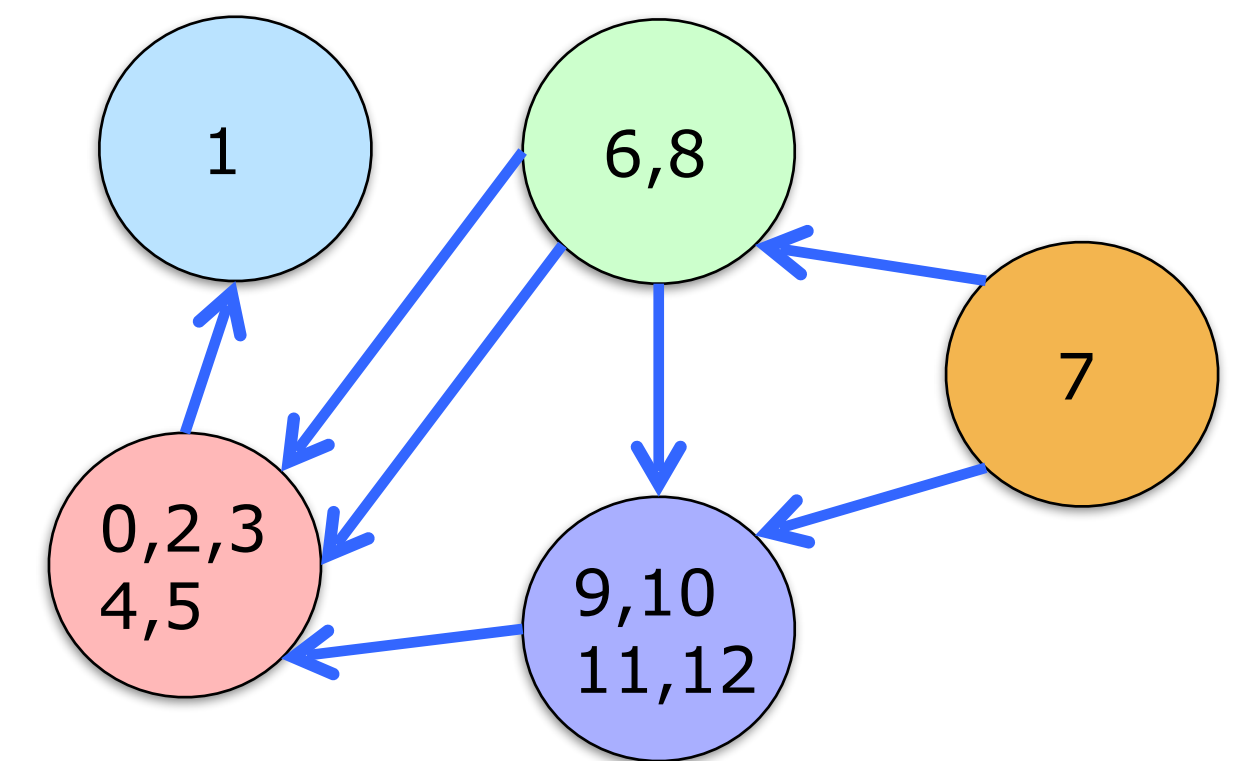
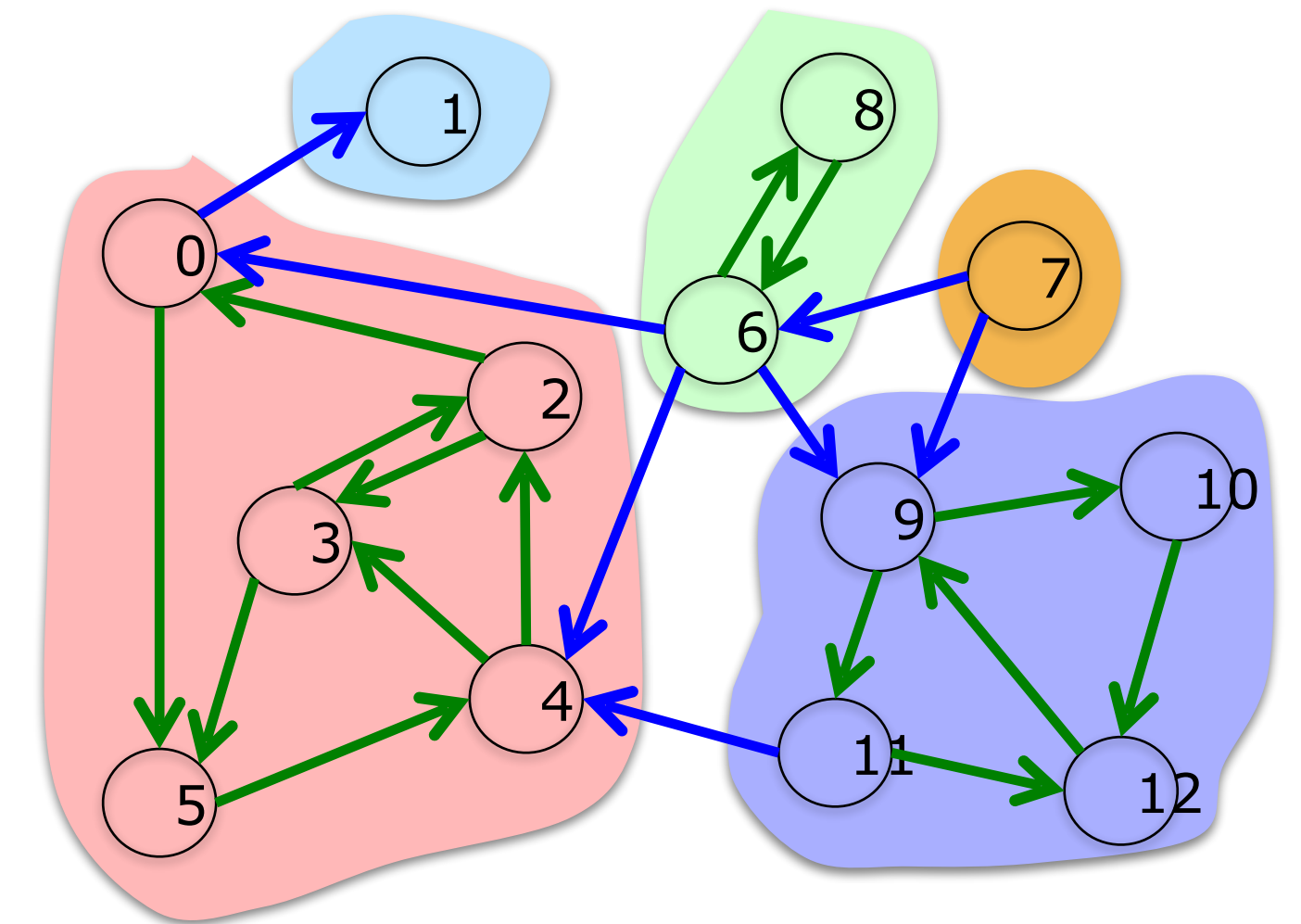
v et w sont fortement connectés s'il y a un chemin allant de v à w **et** un chemin allant de w à v

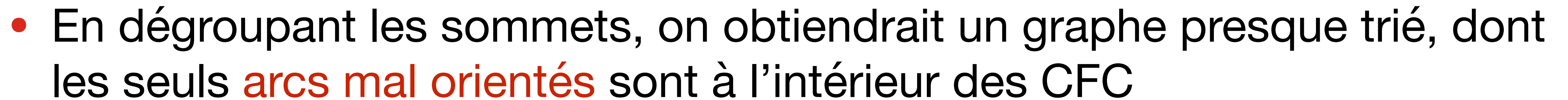


Quelques observations

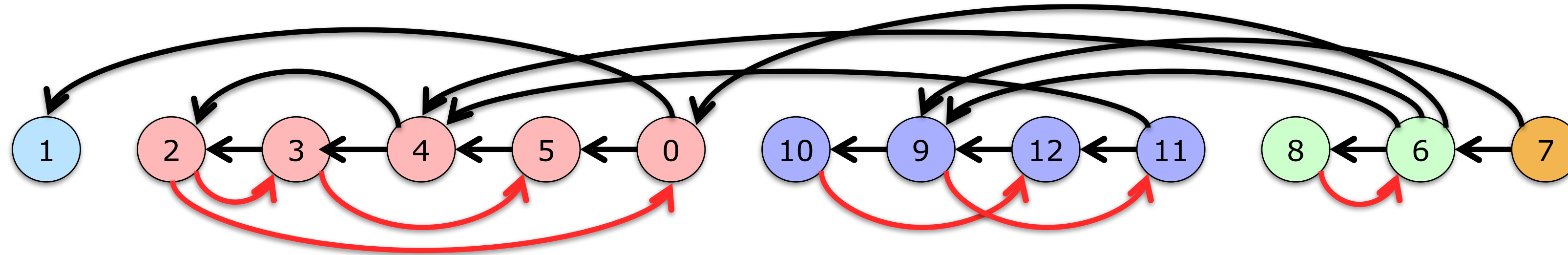
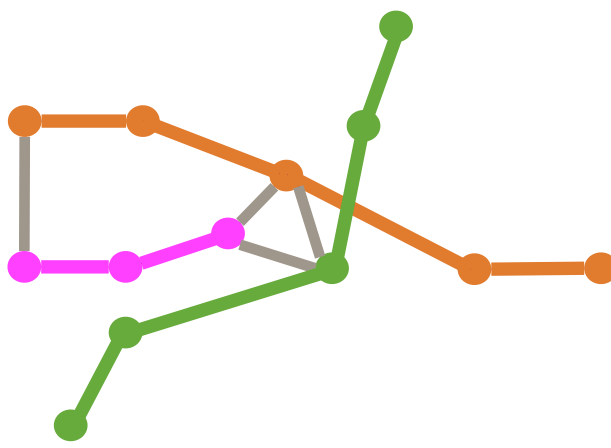


- On distingue deux types d'arcs.
- ceux qui restent dans la même CFC. Ils forment des cycles au sein de chaque CFC.
- ceux qui mènent d'une CFC à une autre.
- En regroupant tous les sommets d'une même CFC et en ne gardant donc que les arcs bleus, on obtient un graphe acyclique





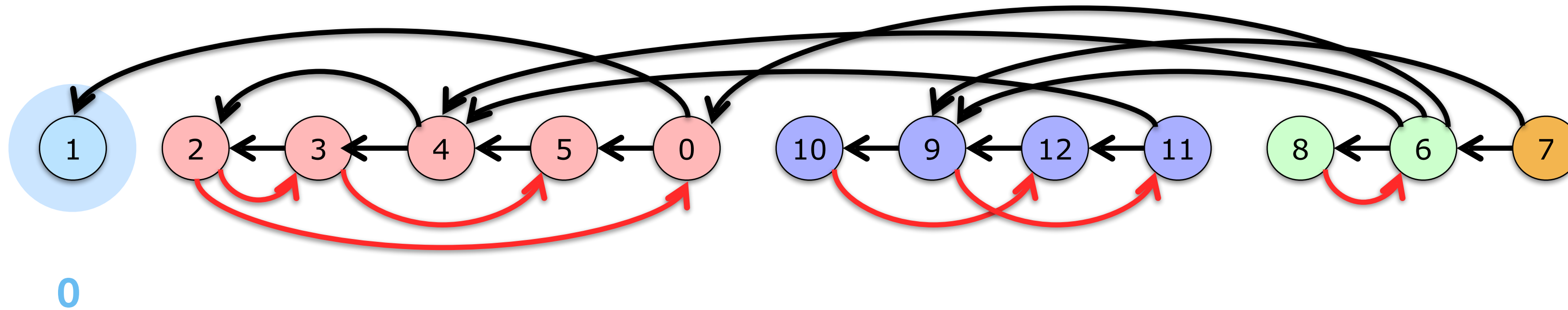
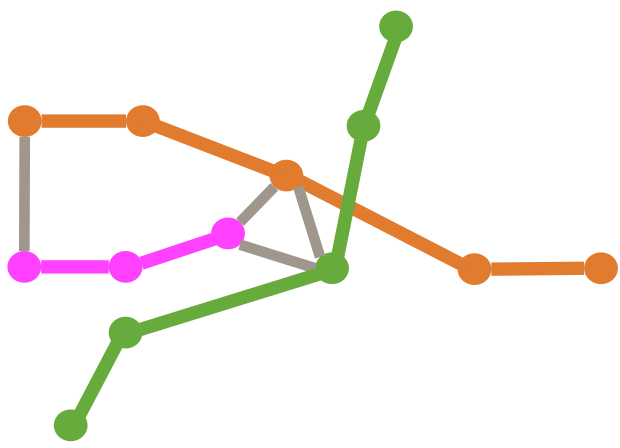
Quelques observations



Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

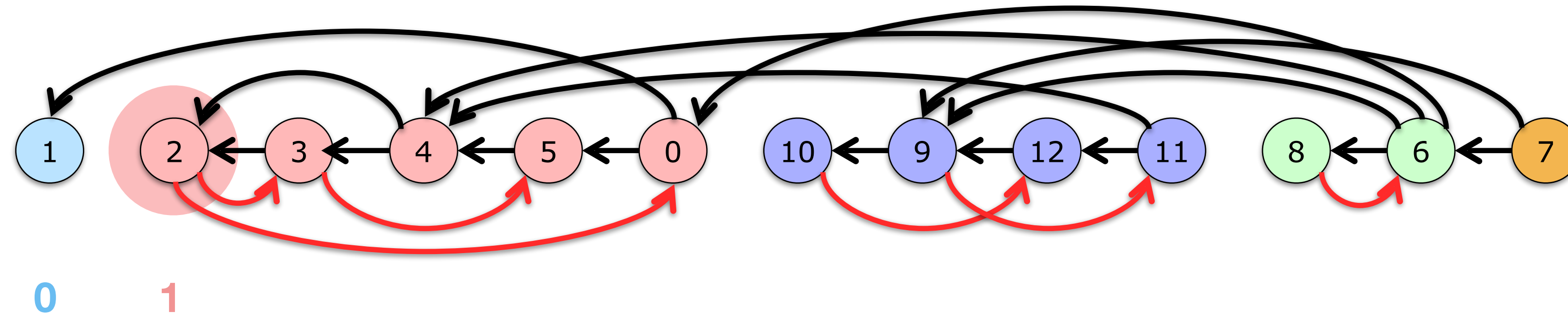
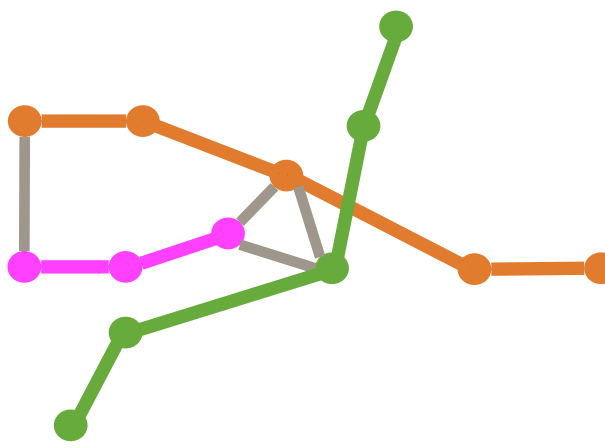


- $\text{DFS}(1) \rightarrow \{ 1 \}$

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

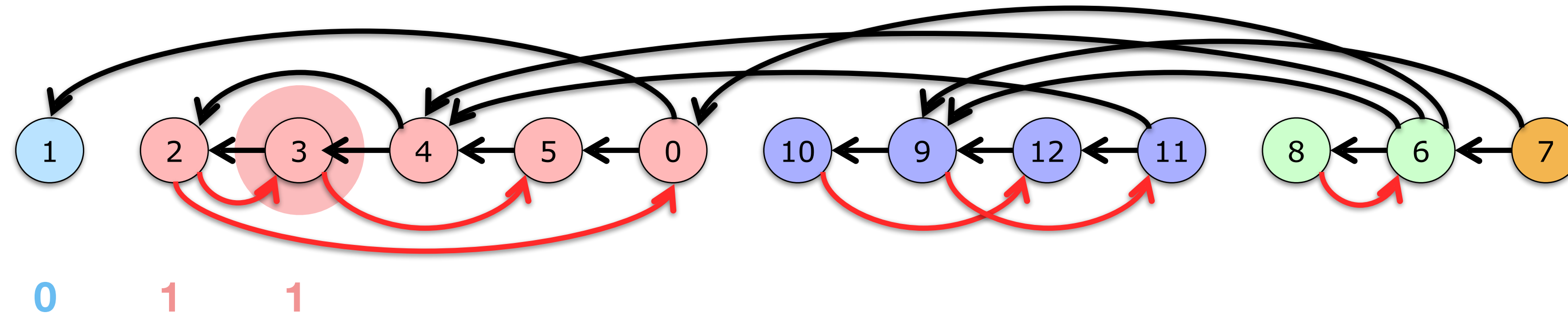
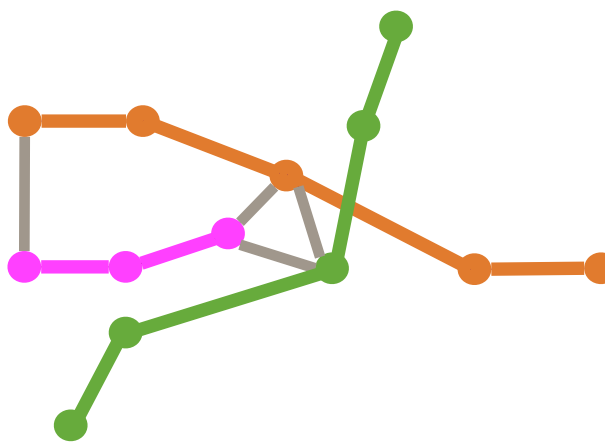


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

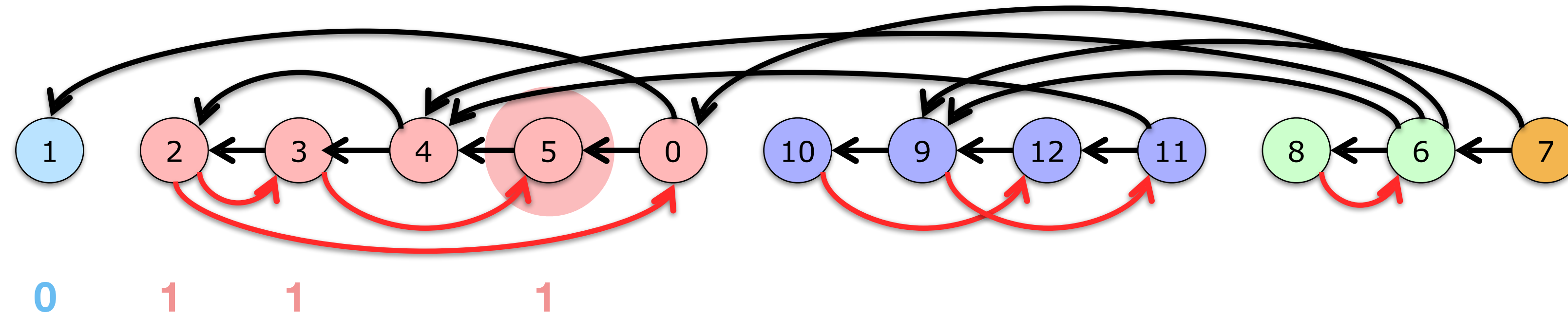
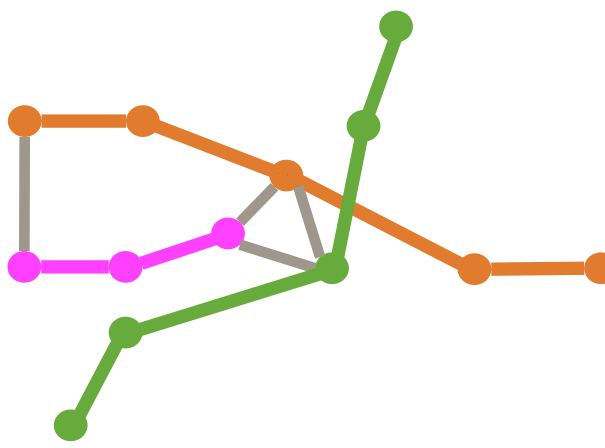


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

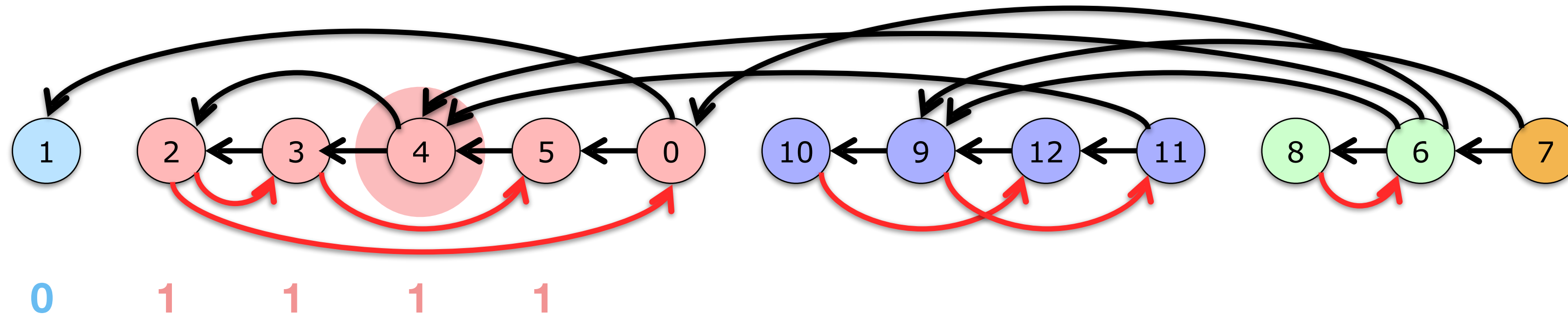
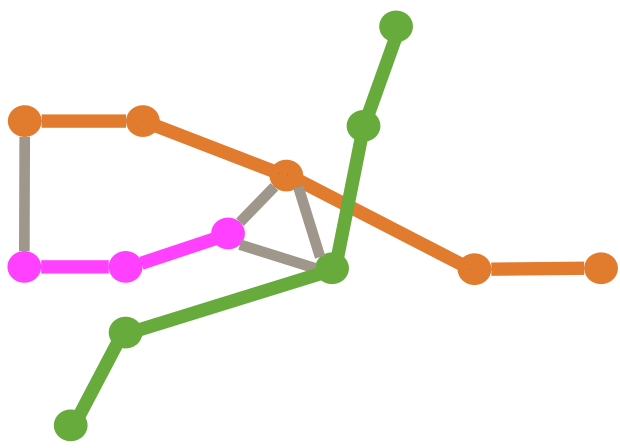


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

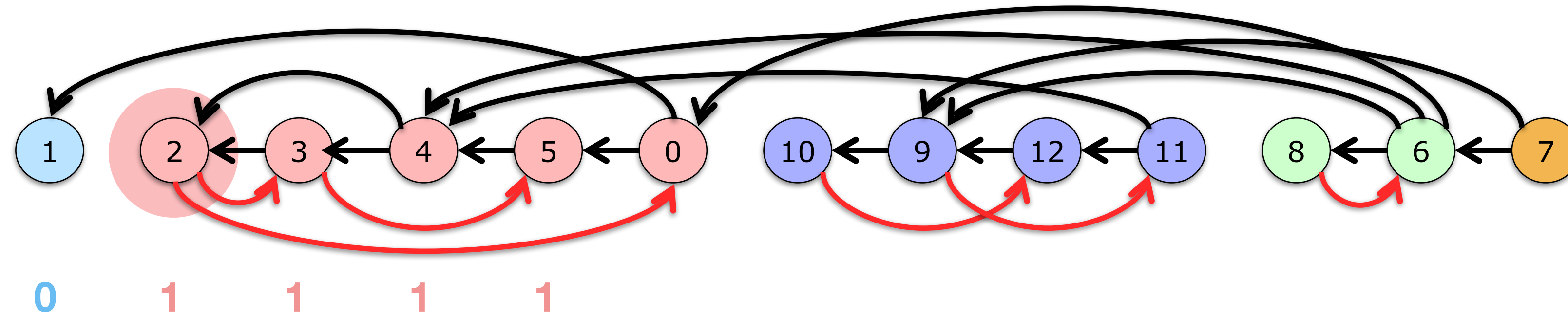
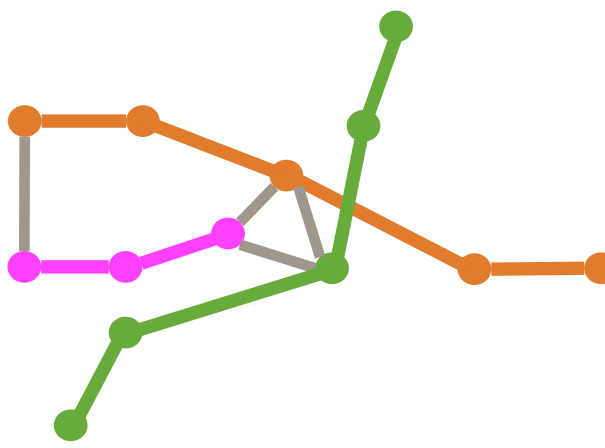


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

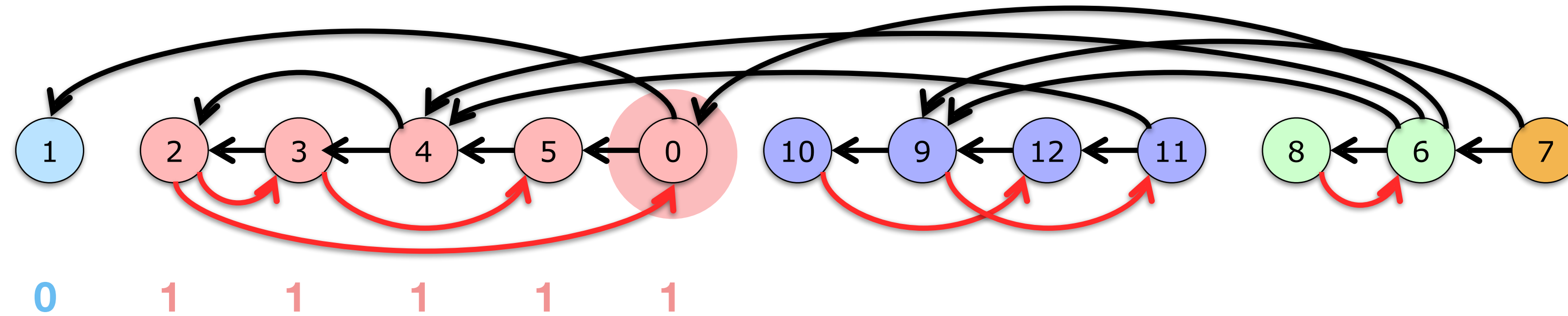
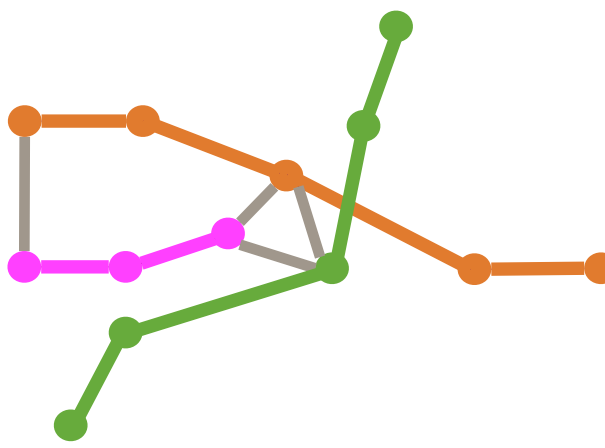


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

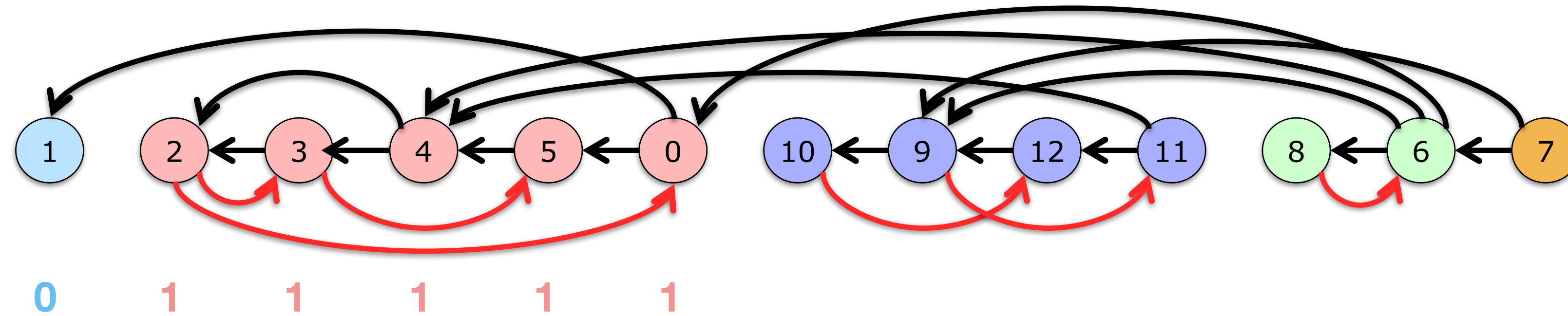
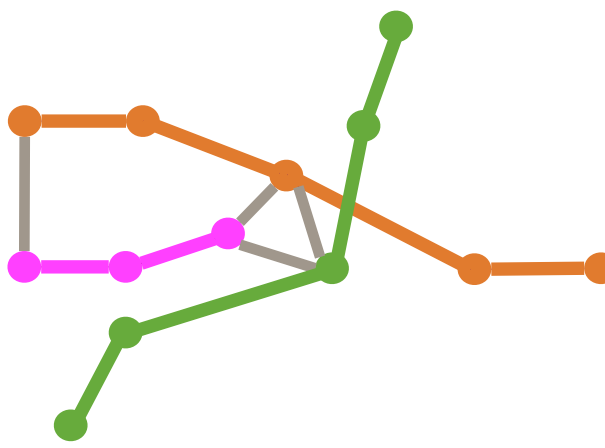


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

Quelques observations

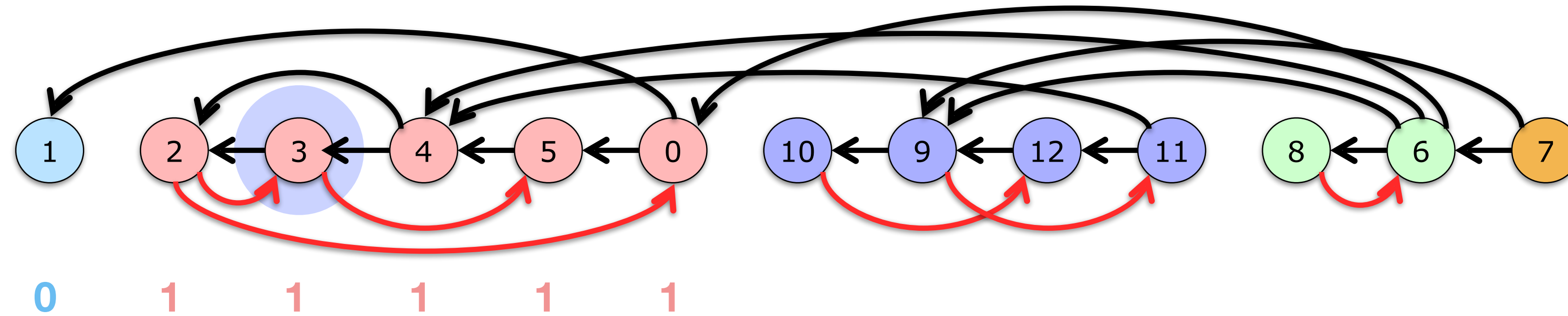
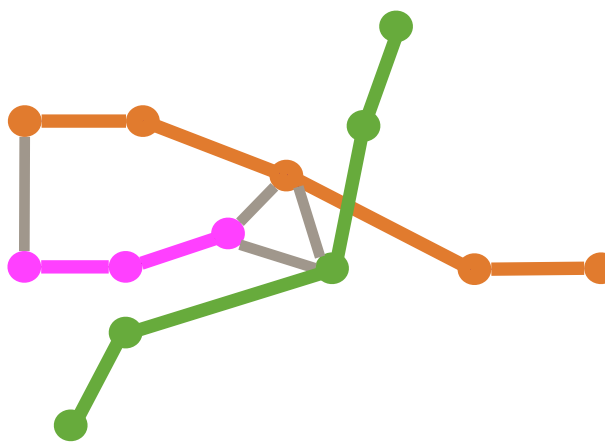


- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment

Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

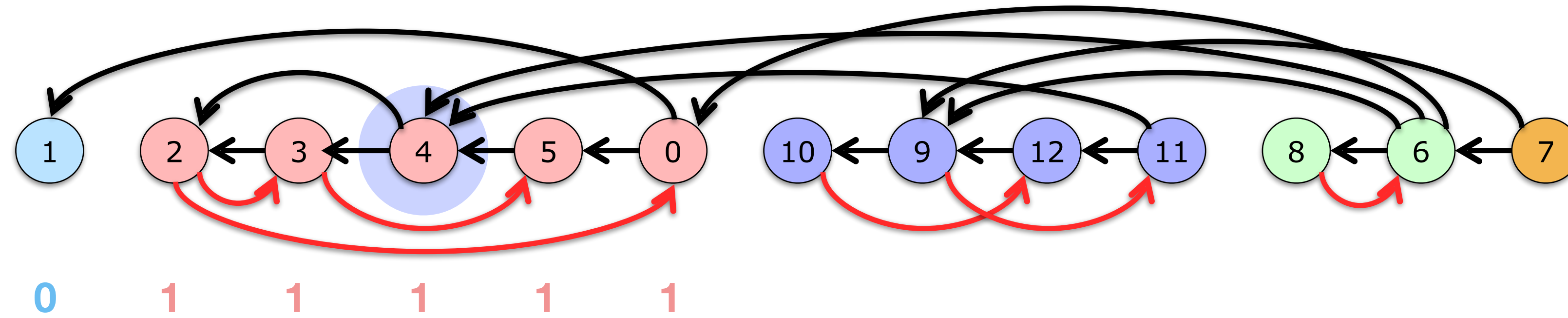
Quelques observations



Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

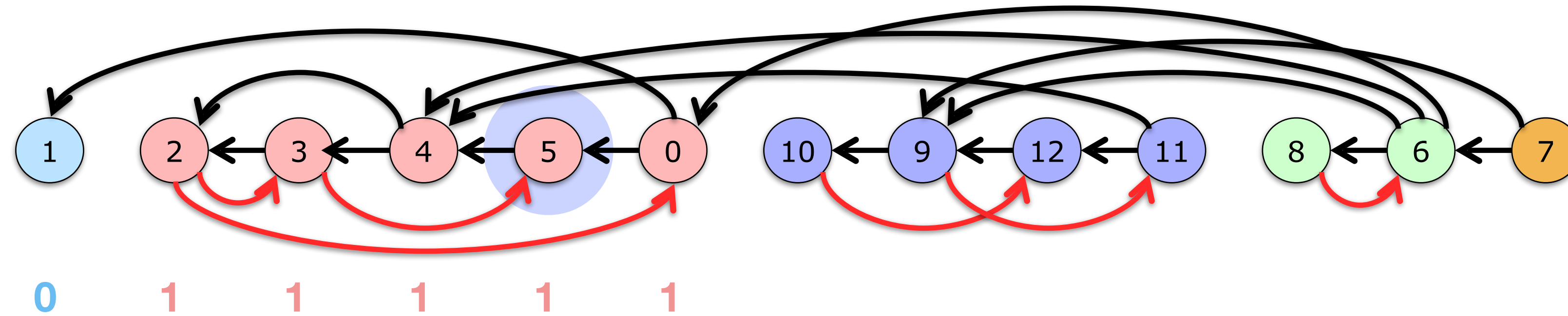
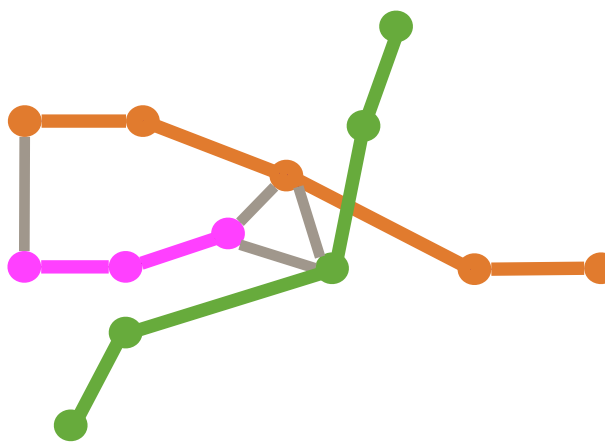
- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$



- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$

ASD - Algorithmes et Structures de Données - Printemps 2021

Quelques observations

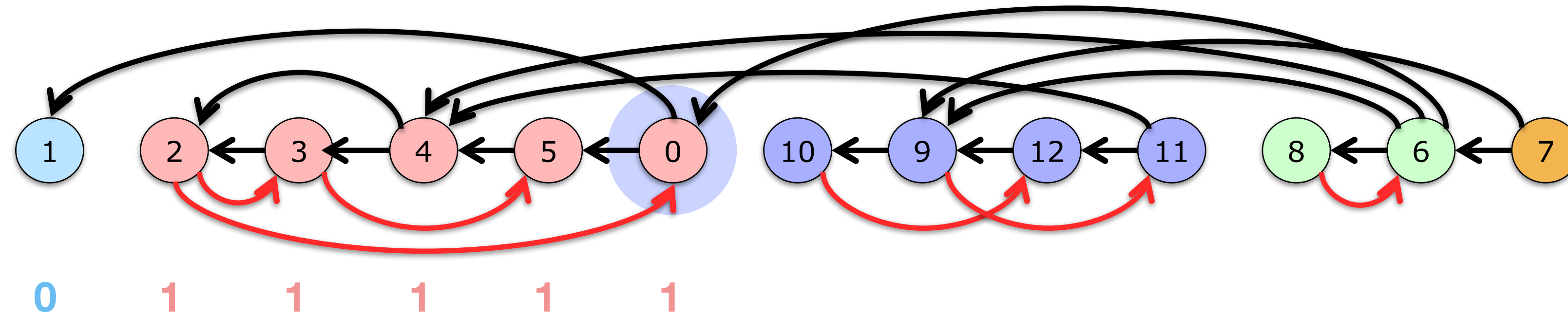
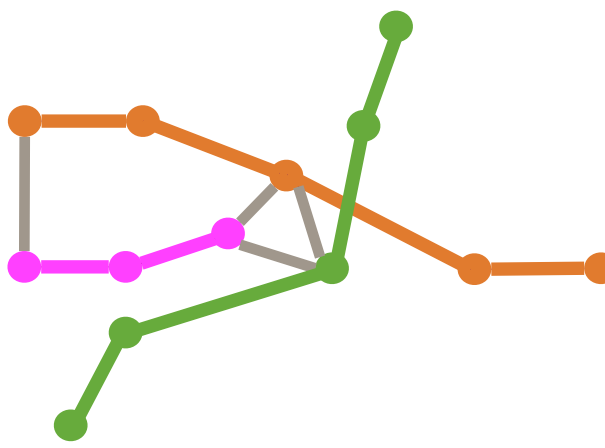


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$

Quelques observations

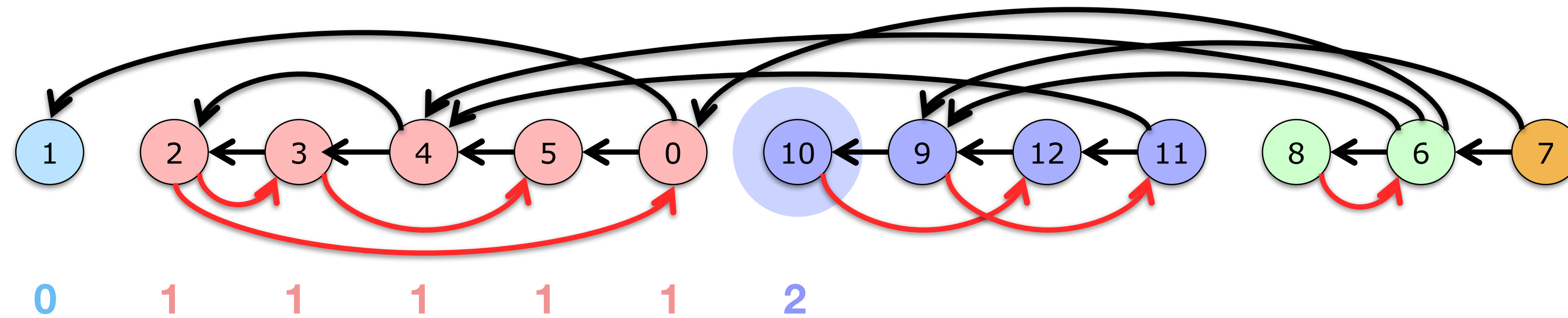
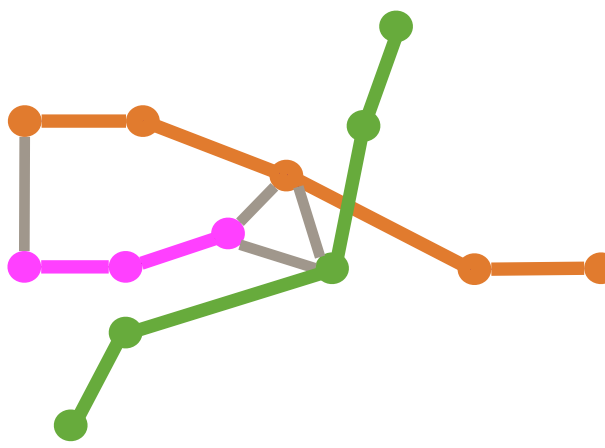


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$

Quelques observations

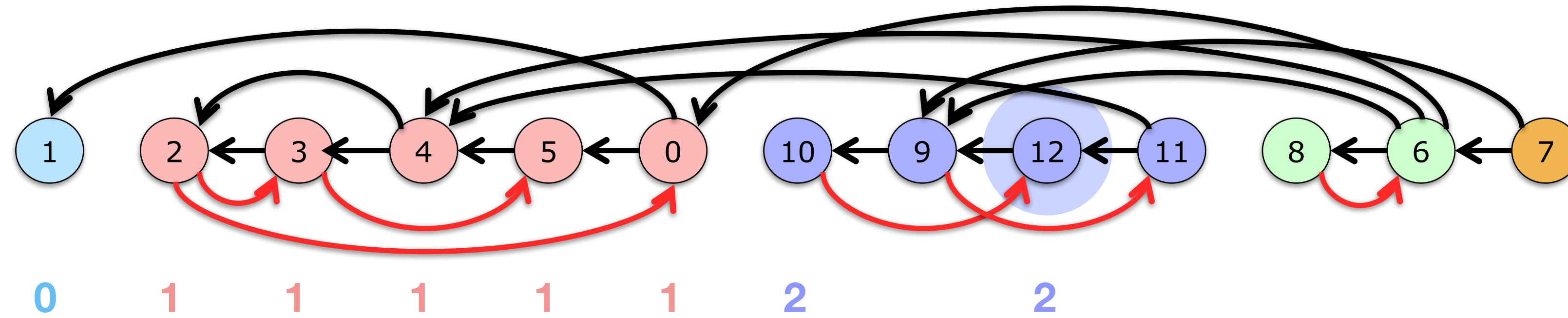
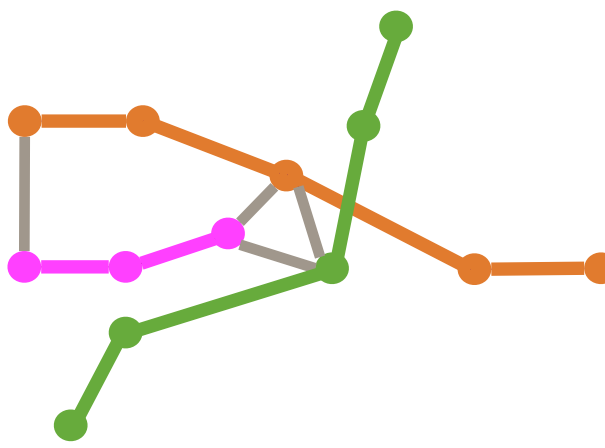


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment

Quelques observations

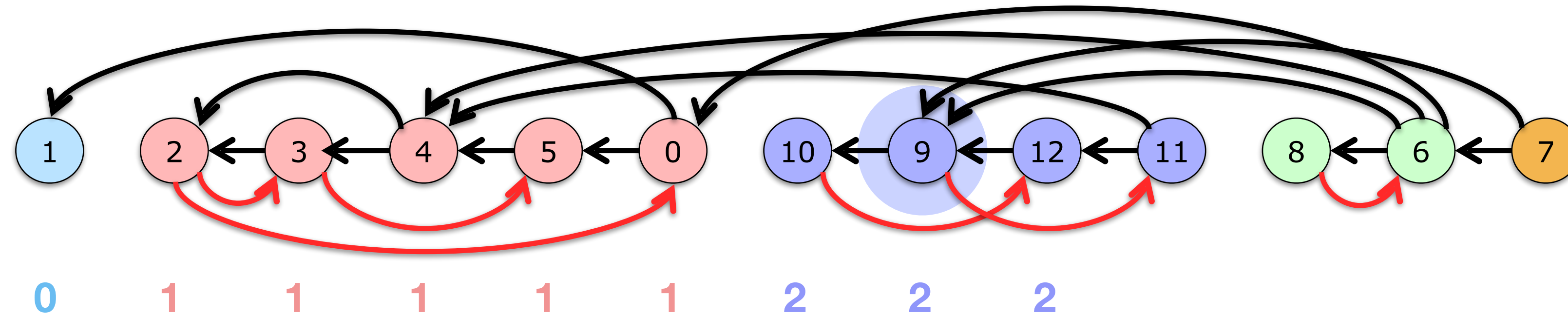
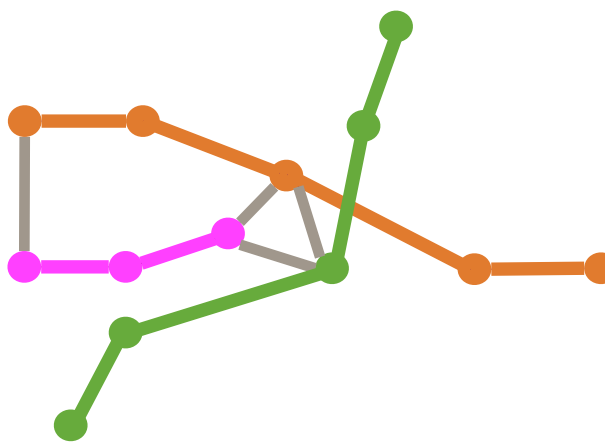


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment

Quelques observations

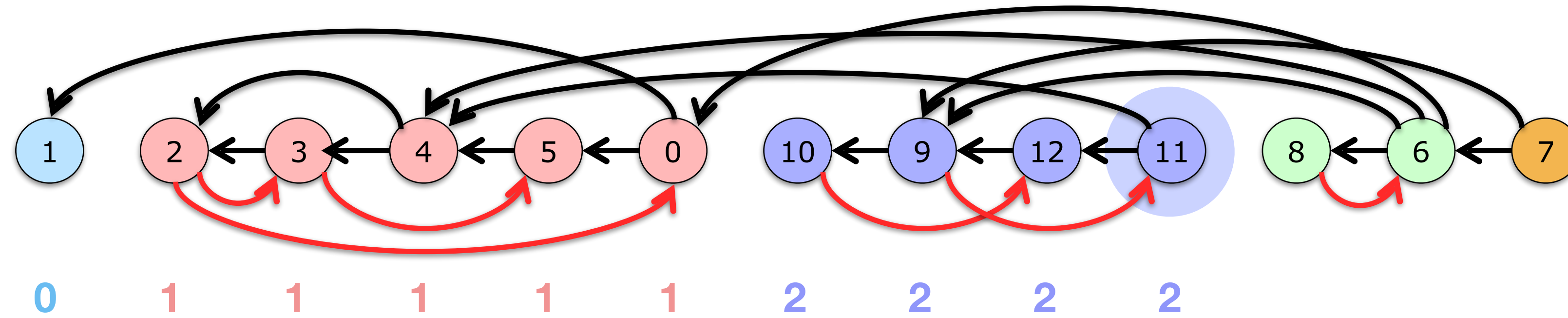
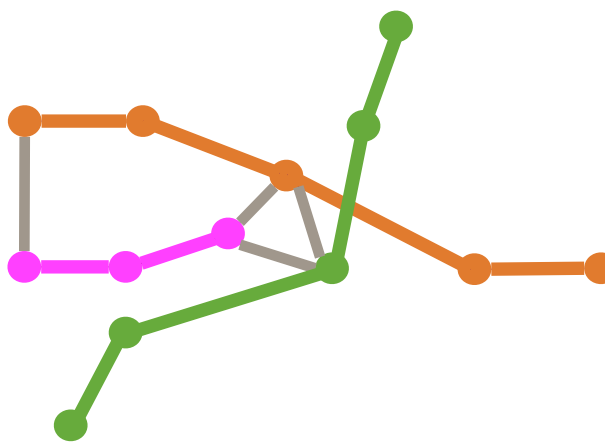


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment

Quelques observations

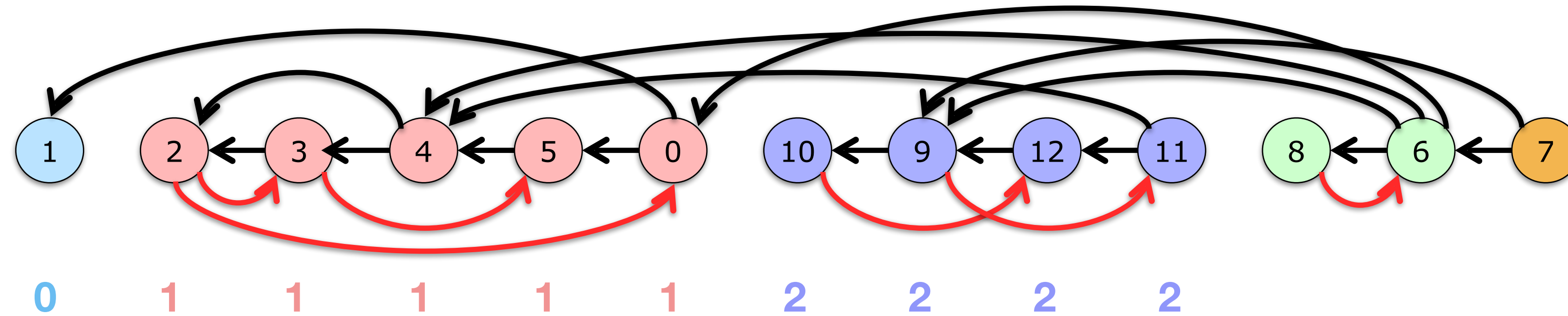
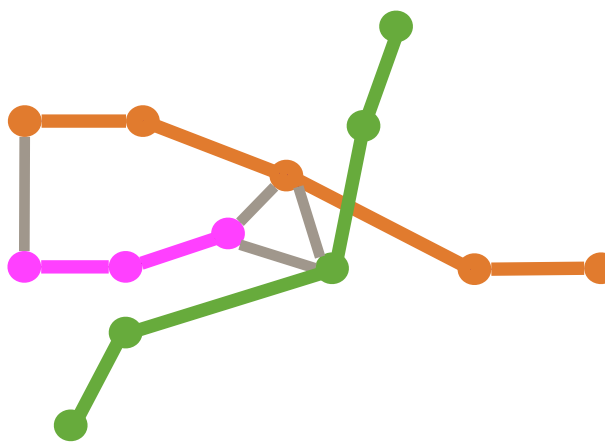


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment

Quelques observations

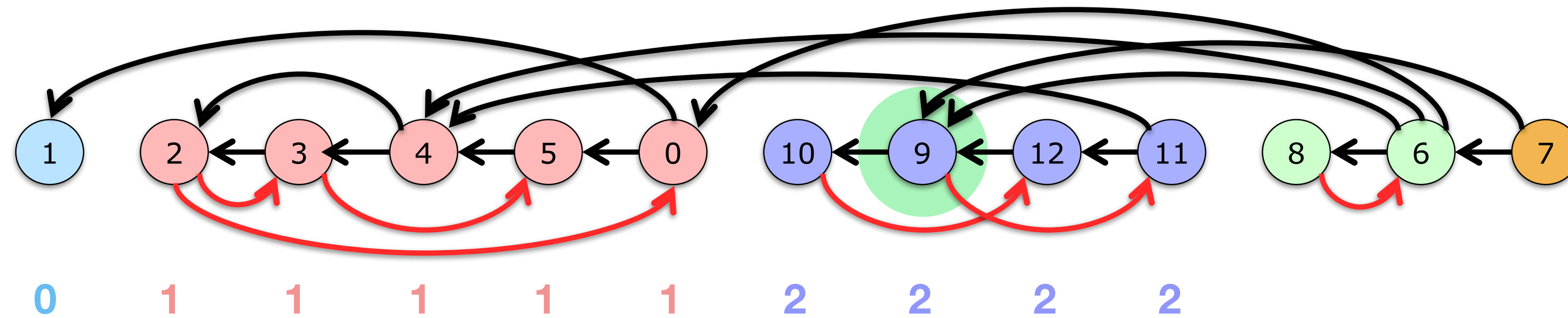
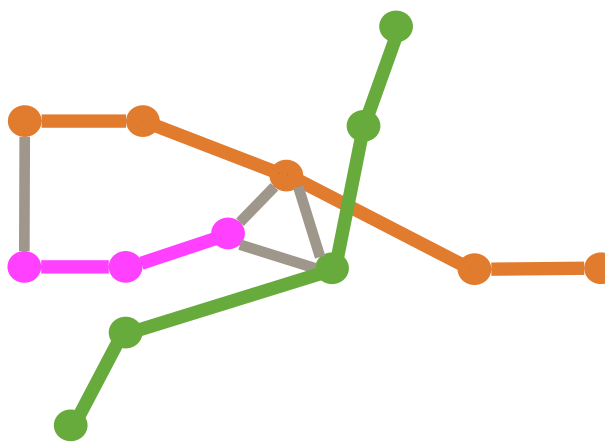


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment

Quelques observations

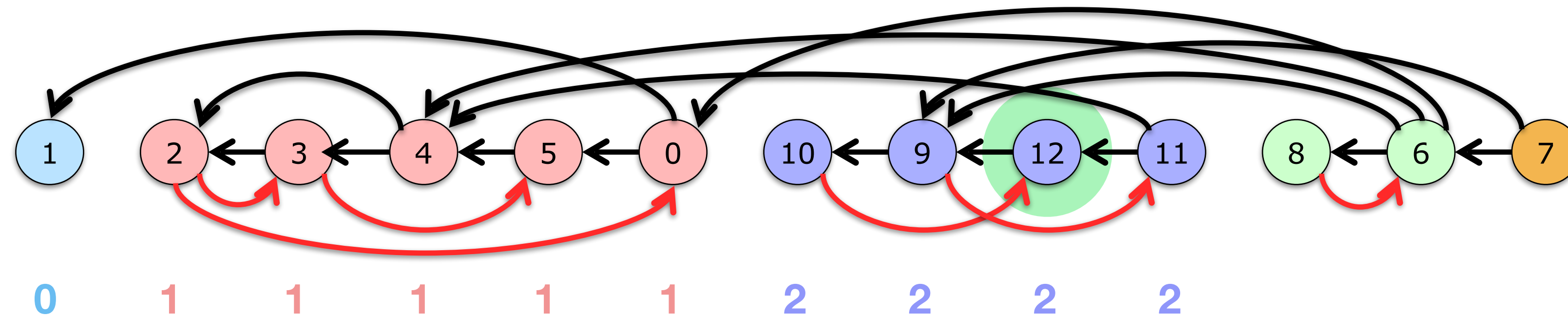
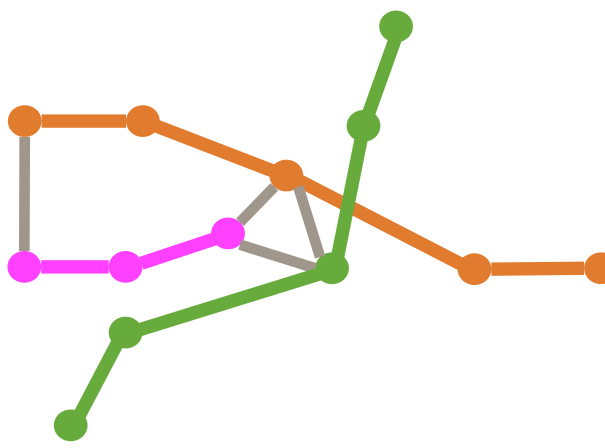


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$

Quelques observations

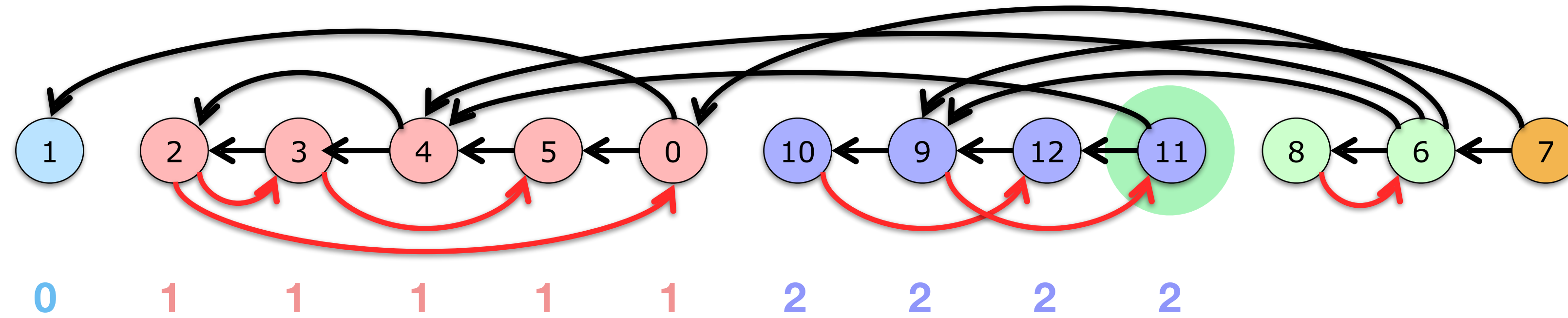
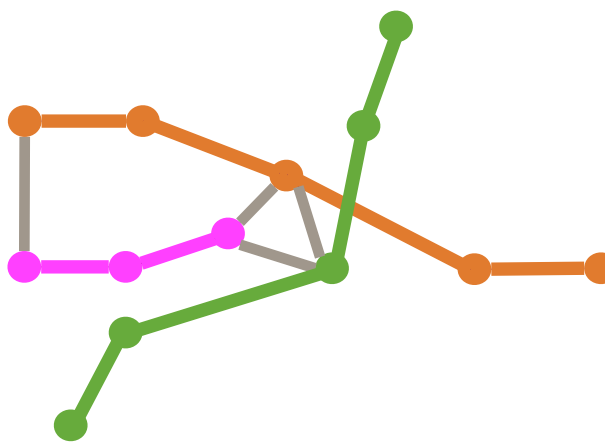


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$

Quelques observations

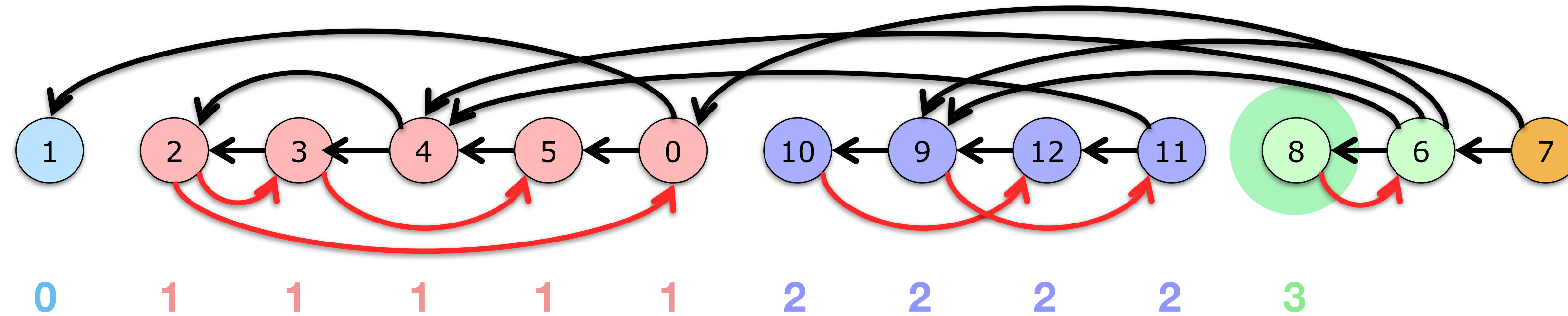
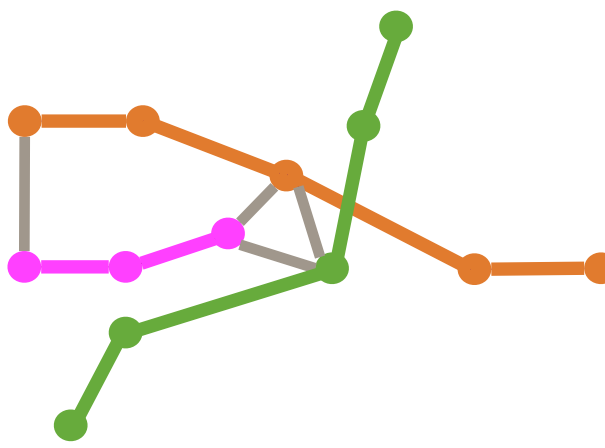


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par DFS(2)
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par DFS(10)

Quelques observations

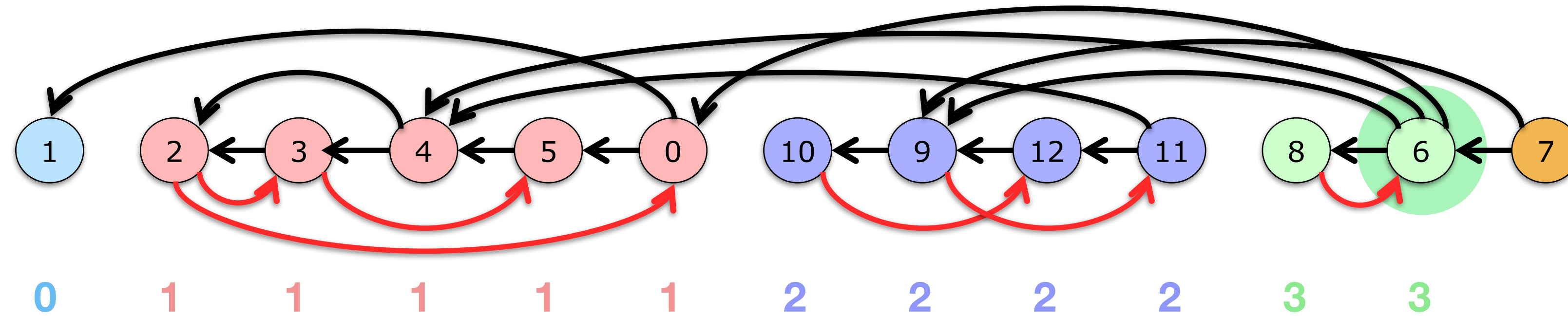
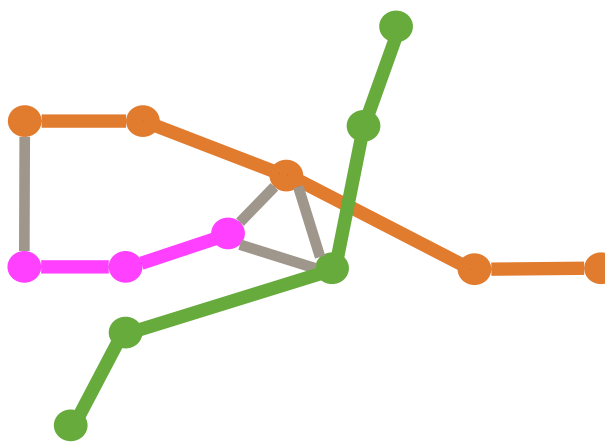


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment

Quelques observations

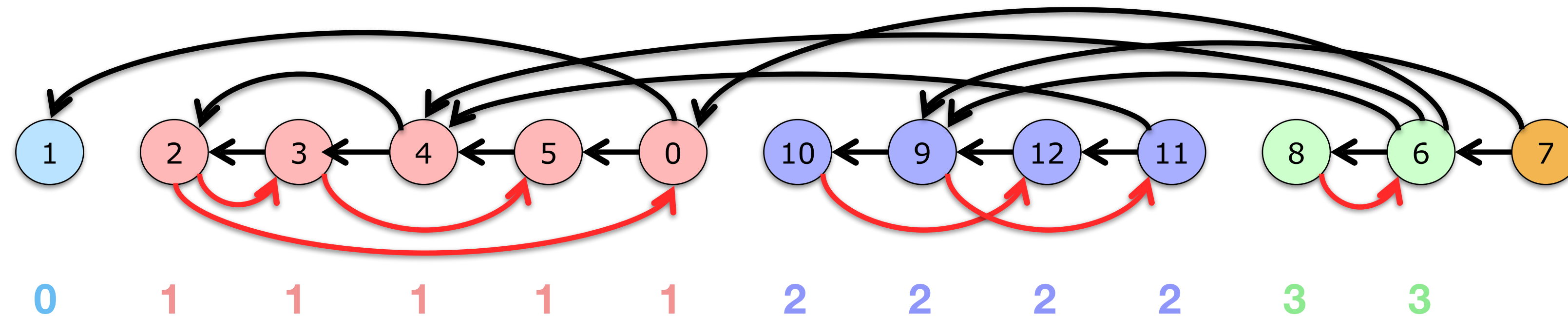
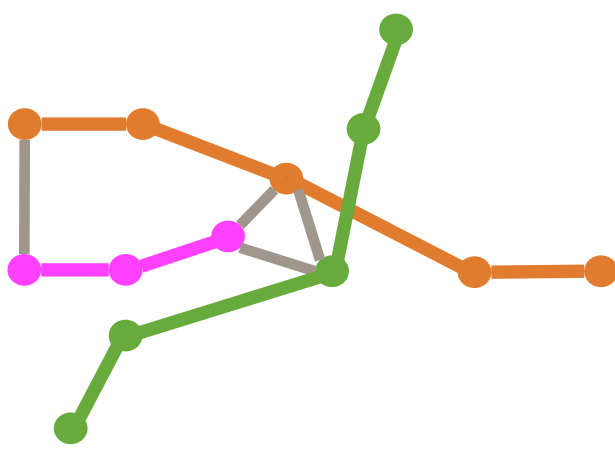


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment

Quelques observations

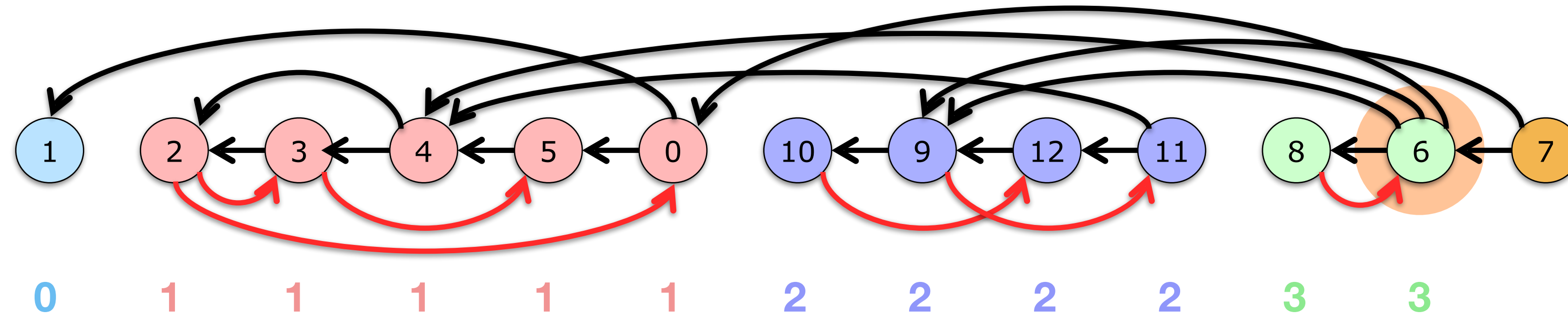
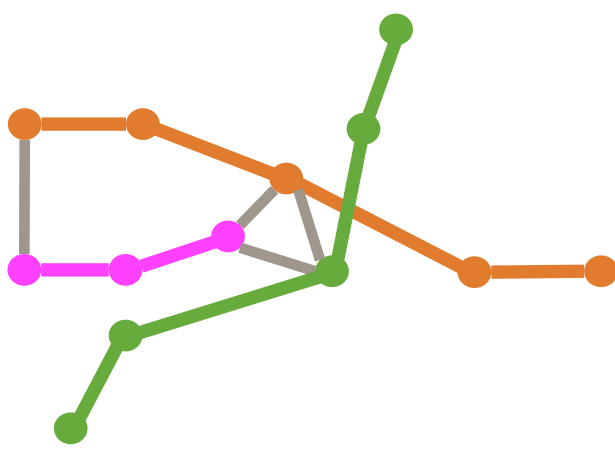


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment

Quelques observations

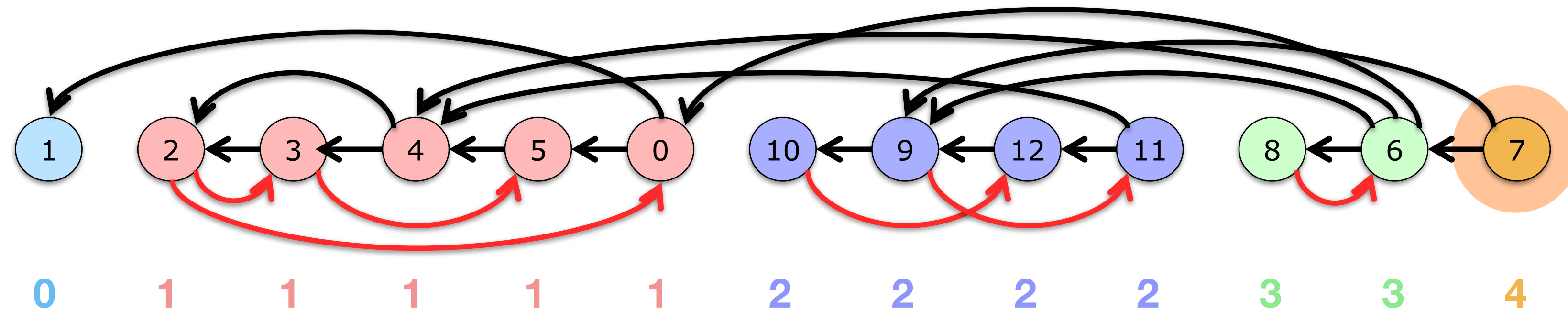
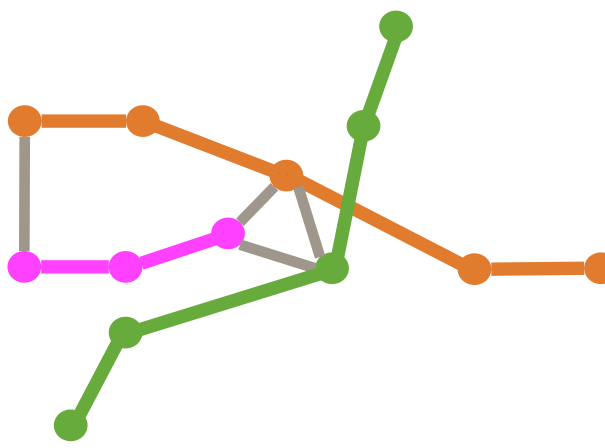


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment
- On passe 6 puisqu'il a été marqué par $\text{DFS}(8)$

Quelques observations

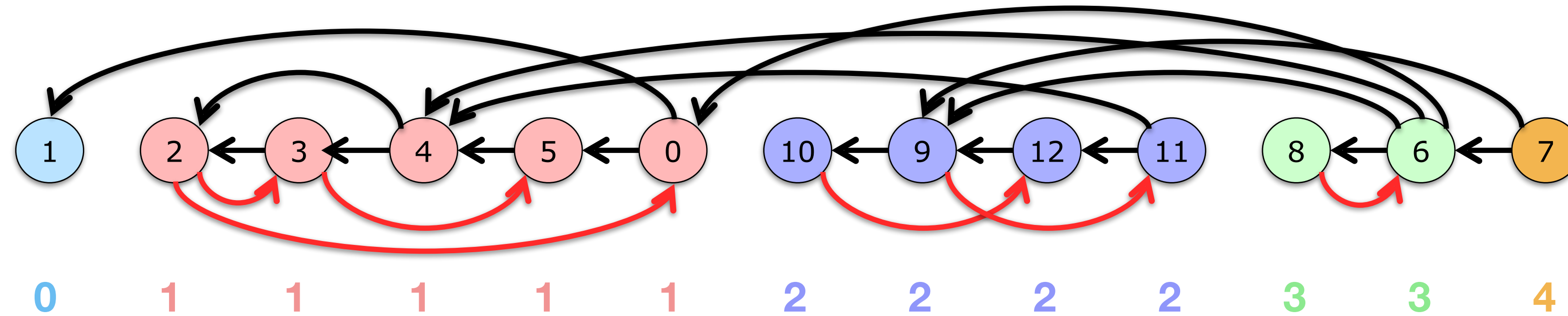
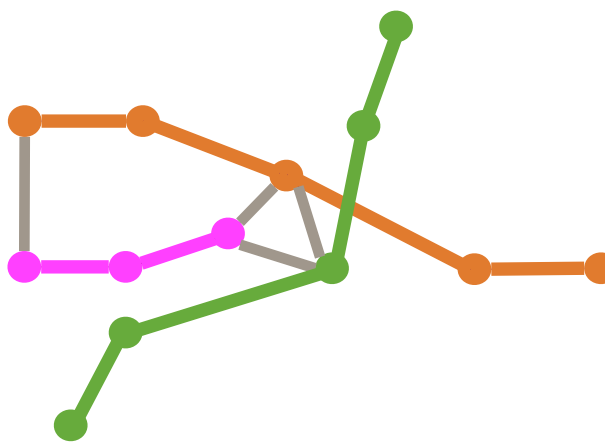


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment
- On passe 6 puisqu'il a été marqué par $\text{DFS}(8)$
- $\text{DFS}(7) \rightarrow \{ 7 \}$ et pas 9 puisque marqué précédemment

Quelques observations

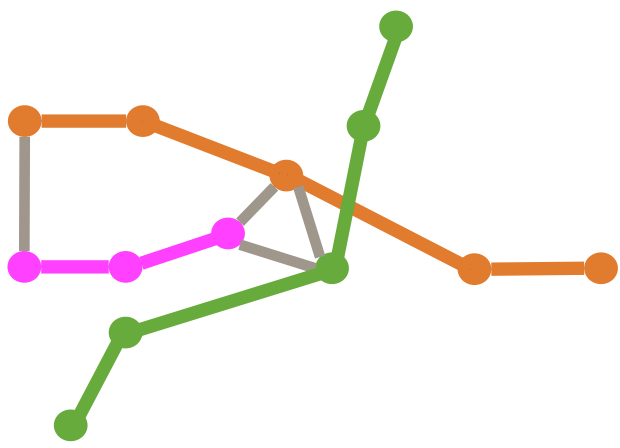


Supposons que les sommets nous sont donnés triés dans cet ordre résultant.

L'algorithme de recherche de CC par parcours en profondeur (DFS) nous donnerait les CFC !

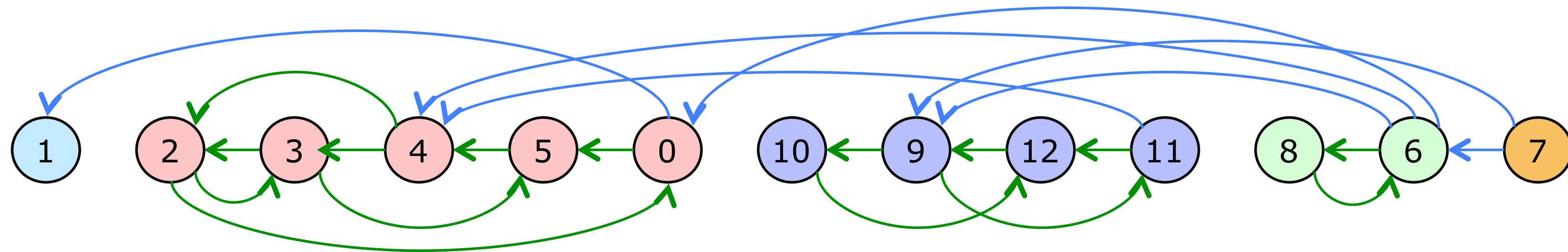
- $\text{DFS}(1) \rightarrow \{ 1 \}$
- $\text{DFS}(2) \rightarrow \{ 2, 3, 5, 4, 0 \}$, et pas 1 puisque marqué précédemment
- On passe 3, 4, 5, 0 puisqu'ils ont été marqués par $\text{DFS}(2)$
- $\text{DFS}(10) \rightarrow \{ 10, 12, 9, 11 \}$, et pas 4 puisque marqué précédemment
- On passe 9, 12, 11 puisqu'ils ont été marqués par $\text{DFS}(10)$
- $\text{DFS}(8) \rightarrow \{ 8, 6 \}$ et pas 0, 4 et 9, puisque marqués précédemment
- On passe 6 puisqu'il a été marqué par $\text{DFS}(8)$
- $\text{DFS}(7) \rightarrow \{ 7 \}$ et pas 9 puisque marqué précédemment

Quelques observations



On ne peut pas trier les sommets aussi bien, mais ce n'est pas nécessaire. La propriété d'ordre dont nous avons besoin est

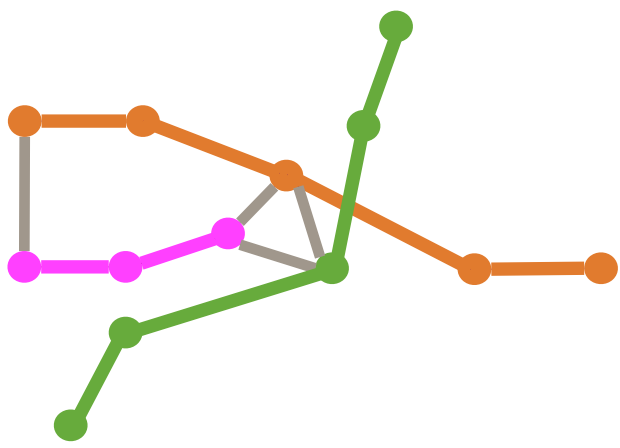
Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC, w apparaît avant tous les sommets de même CFC que v .



(i.e., toutes les flèches bleues pointent vers la gauche)

Cela ressemble beaucoup à un tri topologique, mais inversé, et tolérant les circuits.

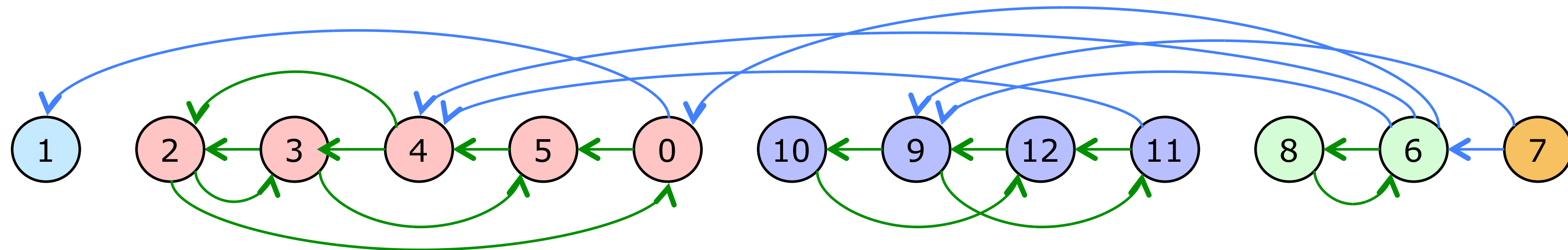
Idée de l'algorithme



Il y a donc deux étapes

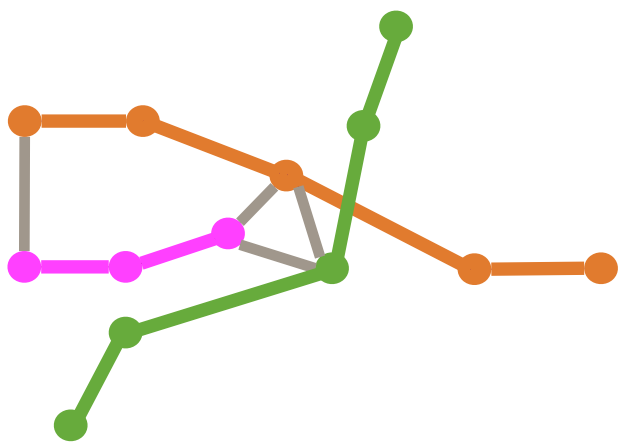
1. Trier les sommets de telle manière à ce que

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparait avant tous les sommets de même CFC que v .



2. Effectuer une recherche de CC par DFS en utilisant cet ordre.

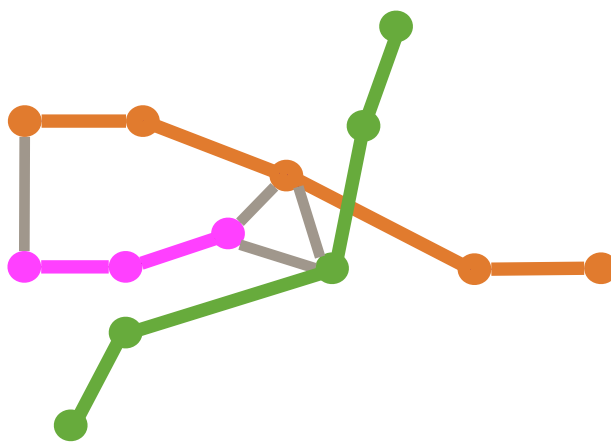
Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Trier les sommets



Propriété souhaitée

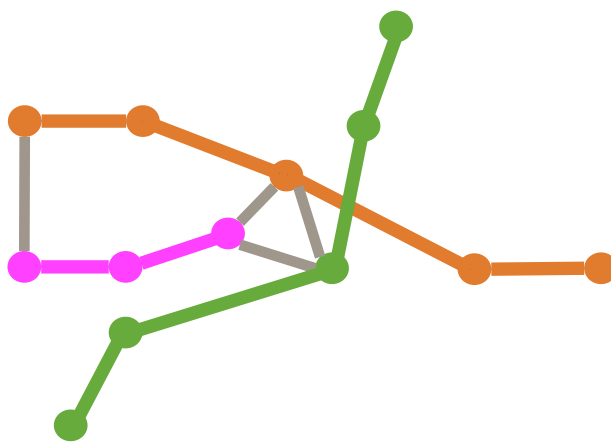
Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Rappel

Le post-ordre inverse sur un graphe contenant des cycles est tel que

*Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à un même cycle,
 v apparaît avant w .*

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Rappel

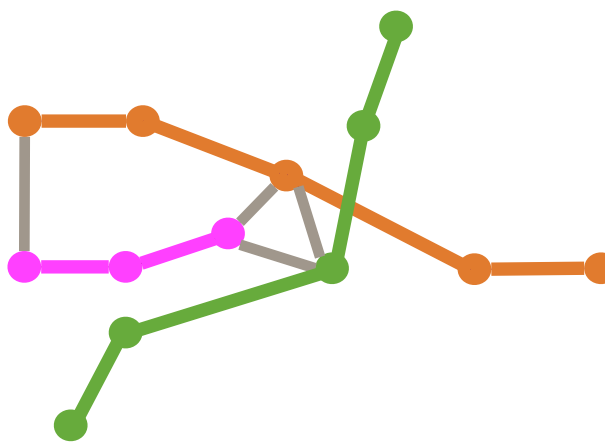
Le post-ordre inverse sur un graphe contenant des cycles est tel que

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à un même cycle,

v apparaît avant w .

(et donc, à une même CFC)

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Rappel

Le post-ordre inverse sur un graphe contenant des cycles est tel que

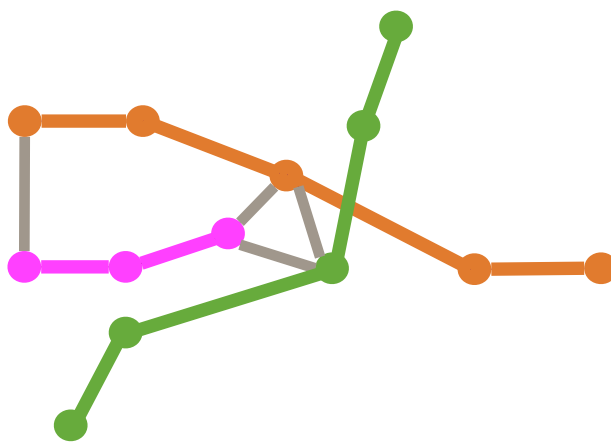
Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à un même cycle,

v apparaît avant w .

(et donc, à une même CFC)

On peut aussi remarquer que tous les sommets appartenant à la même CFC que
 w apparaissent aussi après v .

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC, w apparaît avant tous les sommets de même CFC que v .

Rappel

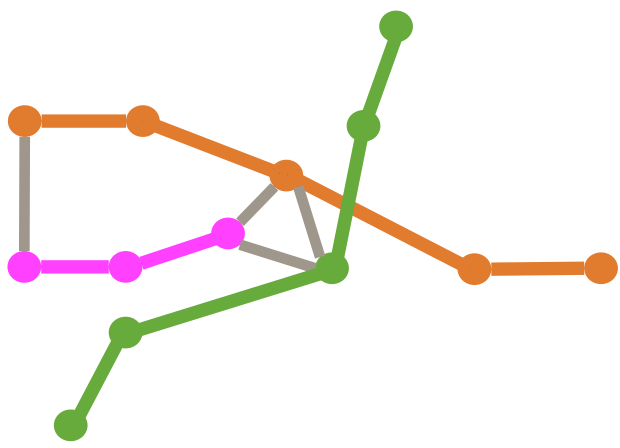
Le post-ordre inverse sur un graphe contenant des cycles est tel que

*Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à un même cycle,
 v apparaît avant w .*
(et donc, à une même CFC)

On peut aussi remarquer que tous les sommets appartenant à la même CFC que w apparaissent aussi après v .

En effet, si v et w appartiennent à deux CFC différentes, et w est affiché avant v , alors DFS(w) a aussi parcouru (et donc affiché) tous les sommets de même CFC que w avant de terminer.

Trier les sommets



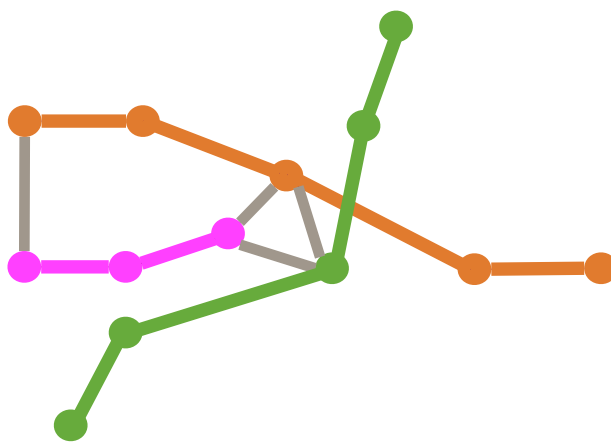
Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Rappel

On peut donc reformuler en disant :

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

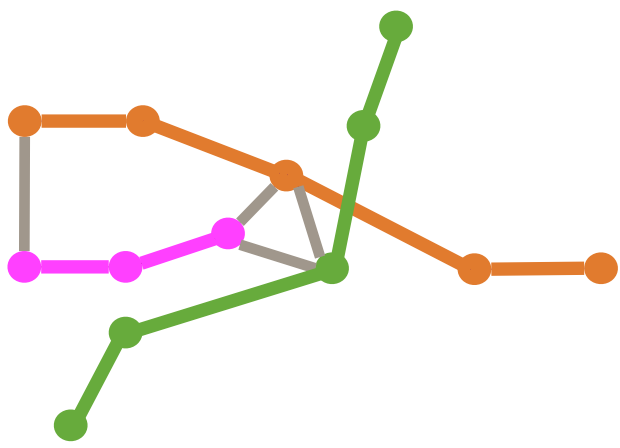
Rappel

On peut donc reformuler en disant :

Le post-ordre inverse sur un graphe contenant des cycles est tel que

*Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 v apparaît avant tous les sommets de même CFC que w .*

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

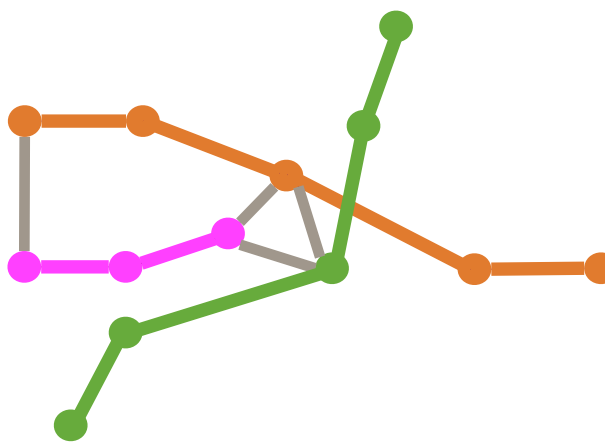
Rappel

On peut donc reformuler en disant :

Le post-ordre inverse sur un graphe contenant des cycles est tel que

*Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 v apparaît avant tous les sommets de même CFC que w .*

Trier les sommets



Propriété souhaitée

Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 w apparaît avant tous les sommets de même CFC que v .

Rappel

On peut donc reformuler en disant :

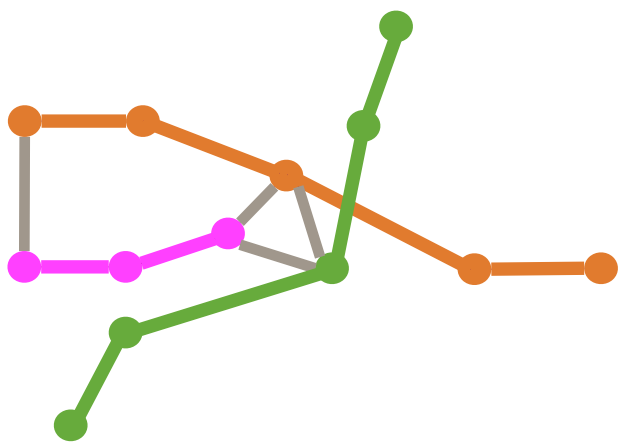
Le post-ordre inverse sur un graphe contenant des cycles est tel que

*Pour tout $v \rightarrow w$ tel que v et w n'appartiennent pas à une même CFC,
 v apparaît avant tous les sommets de même CFC que w .*

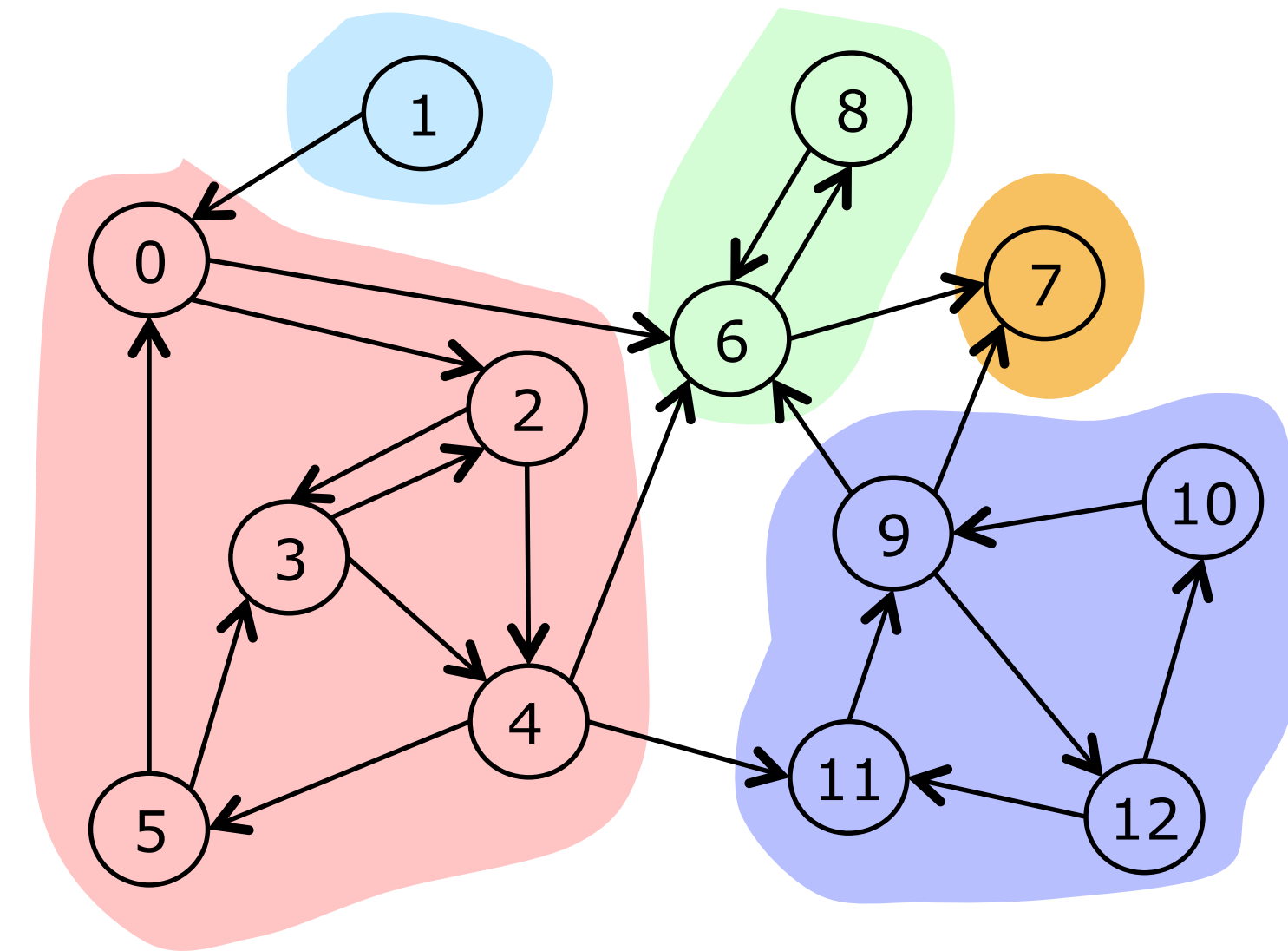
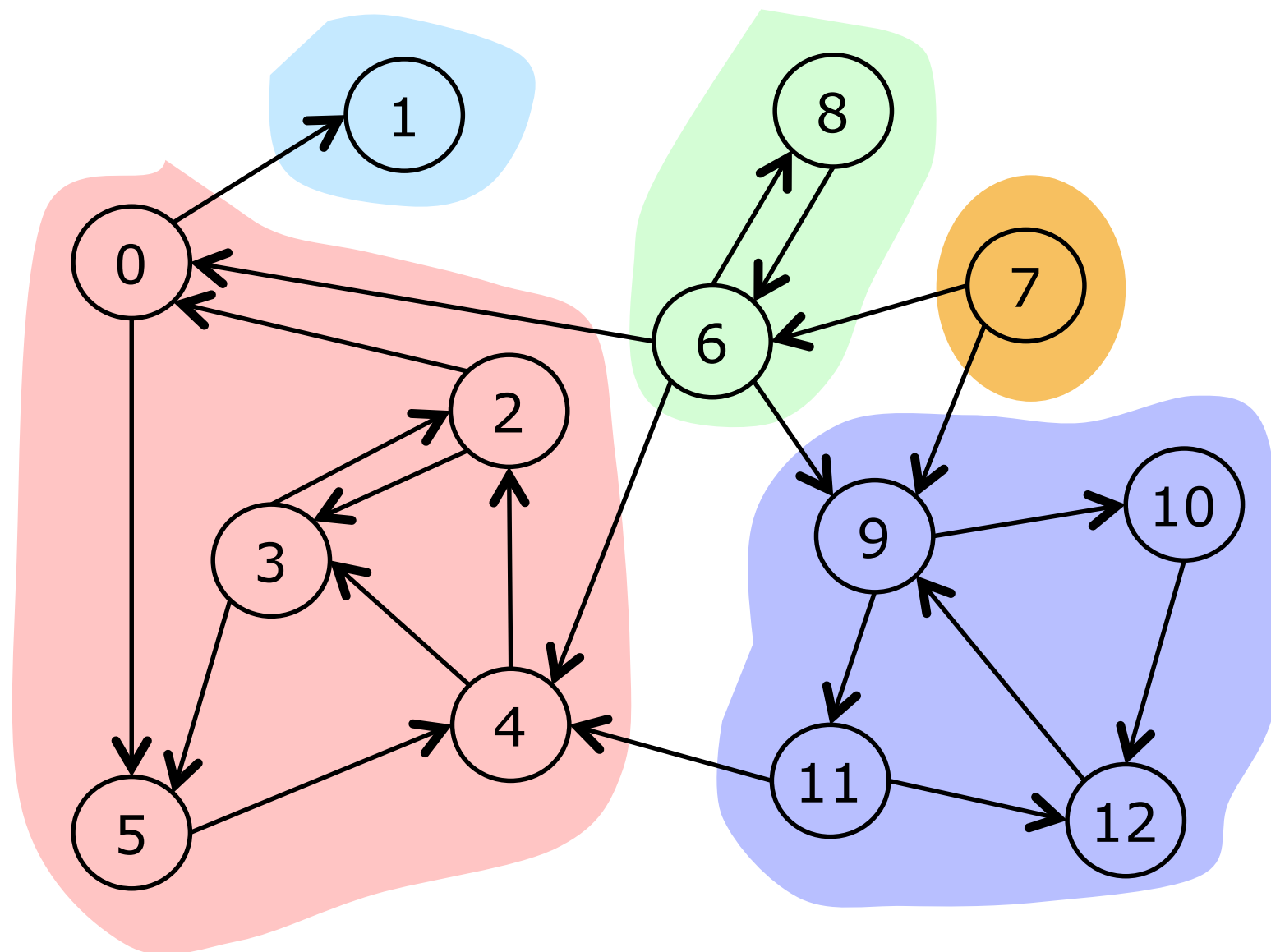
Conclusion

On peut obtenir l'ordre souhaité en faisant un parcours post-ordre inverse sur un graphe dont on a inversé les directions des arcs.

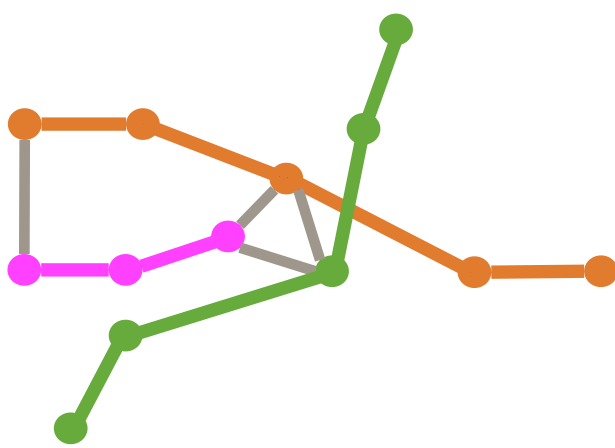
Composantes fortement connexes du graphe inversé



- **Propriété** - Un graphe G et son graphe inverse (*i.e. avec sens de ses arcs inversé*) ont les mêmes composantes fortement connexes

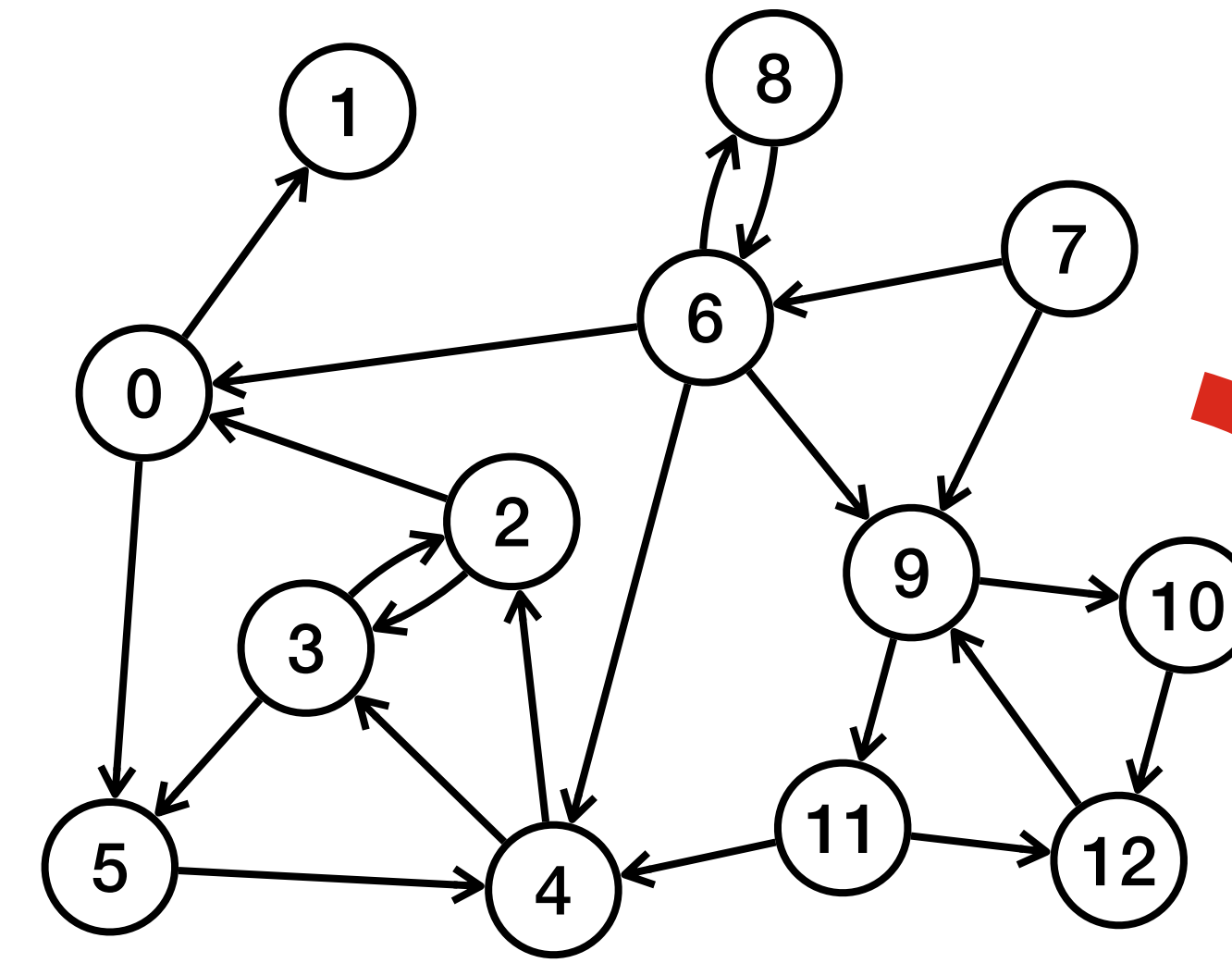


Algorithme de Kosaraju-Sharir



1. Calculer l'inverse du post-ordre DFS du graphe inverse de G .

Pre Ordre:	0	2	3	4	5	11	9	6	7	8	12	10	1
Post Ordre:	5	7	8	6	10	12	9	11	4	3	2	0	1
Post Inverse:	1	0	2	3	4	11	9	12	10	6	8	7	5

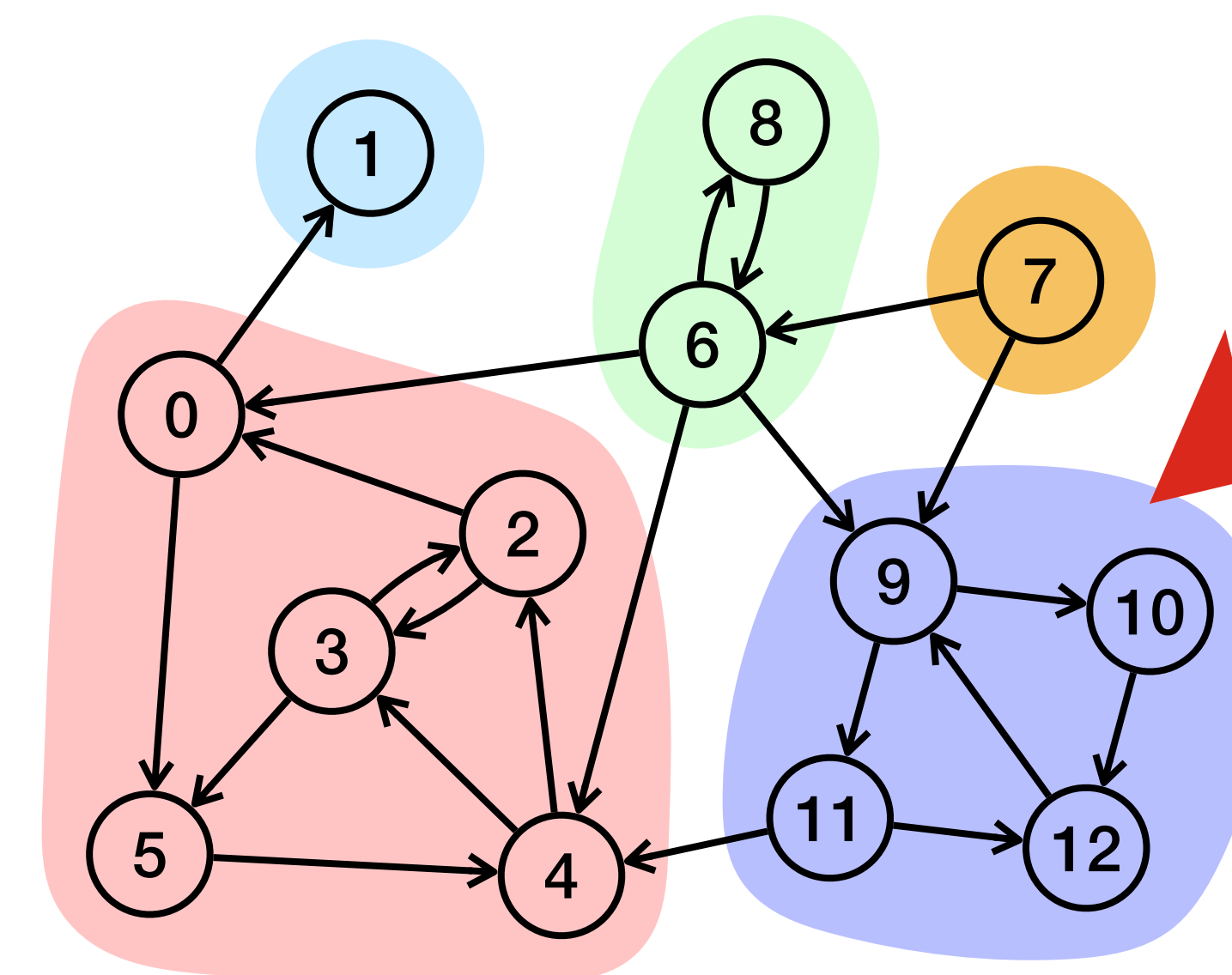


*Post-ordre DFS
sur G inverse
+ inversion.*

1 0 2 3 4 11 9 12 10 6 8 7 5

2. Effectuer une recherche de CC par DFS en utilisant cet ordre.

DFS(1) ->	{ 1 }
DFS(0) ->	{ 0, 5, 4, 2, 3 }
DFS(11) ->	{ 11, 12, 9, 10 }
DFS(6) ->	{ 6, 8 }
DFS(7) ->	{ 7 }



*Recherche CC
avec cet ordre.*