

## Parcours en profondeur

### Algorithme

---

**Algorithme 1** Parcours en profondeur PP( $G$ )

---

```
1: pour tout  $v \in V(G)$  faire  
2:    $\text{couleur}(v) \leftarrow \text{BLANC}$   
3:    $\text{pere}(v) \leftarrow \text{NIL}$   
4: fin pour  
5:  $\text{temps} \leftarrow 0$   
6: pour tout  $v \in V(G)$  faire  
7:   si  $\text{couleur}(v) = \text{BLANC}$  alors  
8:      $\text{VisiterPP}(v)$   
9:   fin si  
10: fin pour
```

---

---

**Algorithme 2** VisiterPP( $v$ )

---

```
1:  $d(v) \leftarrow \text{temps} \leftarrow \text{temps} + 1$   
2:  $\text{couleur}(v) \leftarrow \text{GRIS}$   
3: pour tout  $w \in \text{Adj}(v)$  faire  
4:   si  $\text{couleur}(w) = \text{BLANC}$  alors  
5:      $\text{pere}(w) \leftarrow v$   
6:      $\text{VisiterPP}(w)$   
7:   fin si  
8: fin pour  
9:  $\text{couleur}(v) \leftarrow \text{NOIR}$   
10:  $f(v) \leftarrow \text{temps} \leftarrow \text{temps} + 1$ 
```

---

## Applications du parcours en profondeur

### Tri topologique

Au cours d'un parcours en profondeur, à chaque fois que l'on noircit un sommet, il est inséré en tête de liste.

A  $PP(G)$ , rajouter une ligne 5 bis :

$liste \leftarrow \{\}$

A  $VisiterPP(v)$ , rajouter  
trois lignes 3.1, 3.2, 3.3

**si**  $couleur(w) = GRIS$  **alors**  
    **retourner** "Existence d'un circuit"  
**fin si**

une ligne 11 :

$INSERER\_TETE(liste, v)$

### Composantes fortement connexes

---

#### Algorithme 3 CFC( $G$ )

---

- 1: Exécuter  $PP(G)$  et trier les sommets selon un ordre décroissant de f
  - 2: Calculer  $G^{-1}$
  - 3: Exécuter  $PP(G^{-1})$
  - 4: Retourner les arborescences obtenues comme composantes fortement connexes de  $G$
-