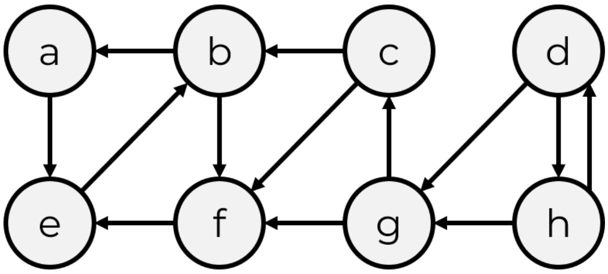
Graphes et algorithmes : TD4

Exercice 1

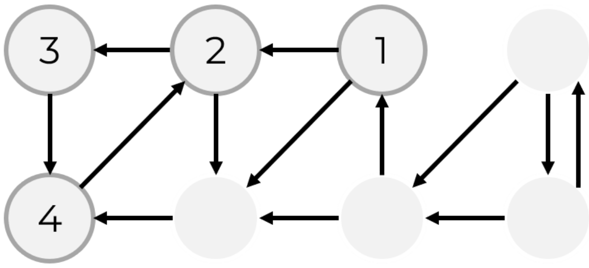
Sujet

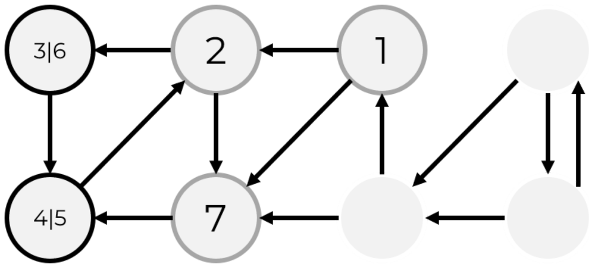
Donner le déroulement de l’algorithme du parcours en profondeur sur le graphe orienté ci-dessous, en prenant pour sommet origine le sommet c. Préciser les dates de découverte et de fin de traitement de chaque sommet.

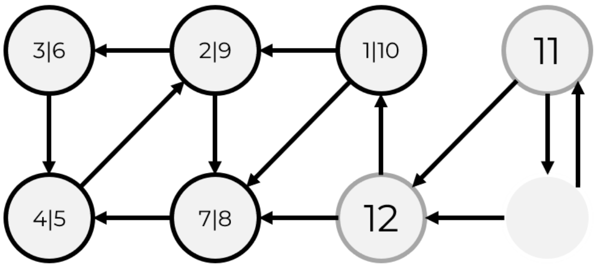


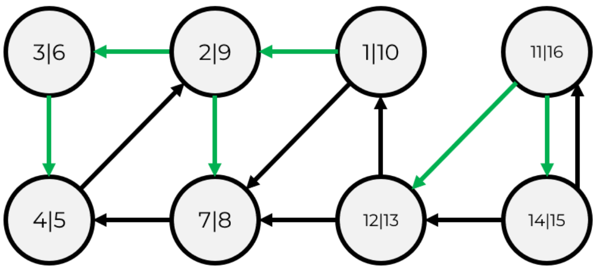
Résolution

- On suppose que les sommets sont traités dans l'ordre alphabétique des identifiants.
    - Nombre à gauche : date de découverte
    - Nombre à droite : date de fin de traitement
    - Contour gris : sommet en cours de traitement
    - Contour noir : sommet déjà traité
    - Contour blanc : sommet non traité
    - Flèche verte : transition utilisée
- On a du relancer l'algorithme à partir du sommet d pour découvrir tout le graphe.





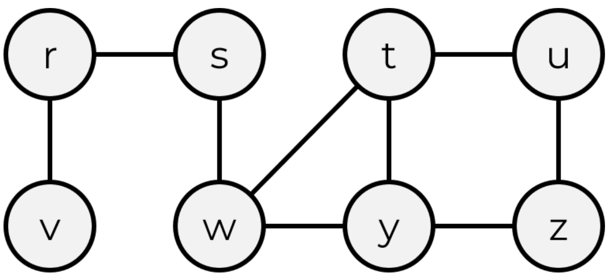




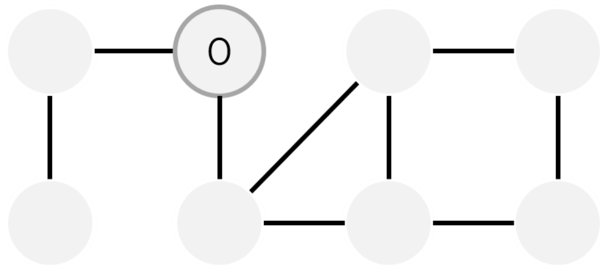
Exercice 3

Sujet

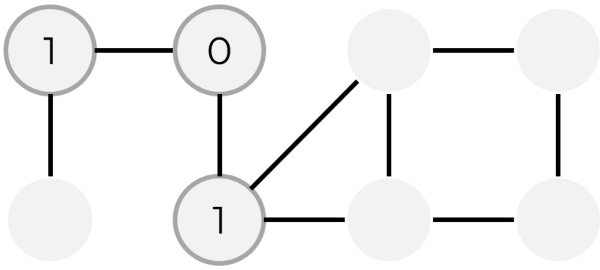
Appliquer l'algorithme de parcours en largeur au graphe non orienté suivant, en partant du sommet s.



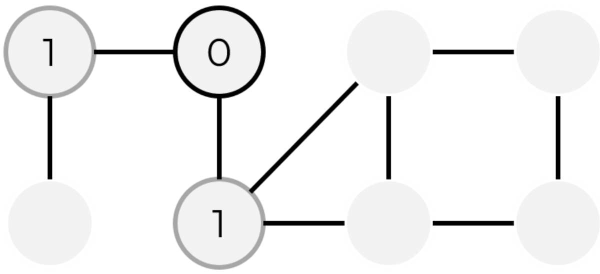
Résolution



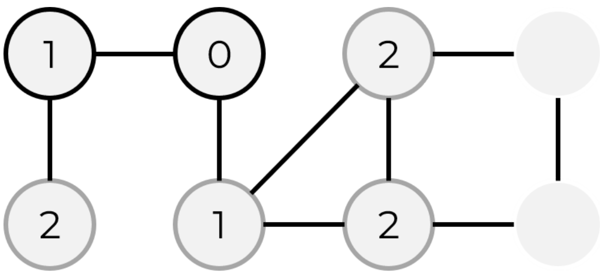
$$F = \{s\}$$



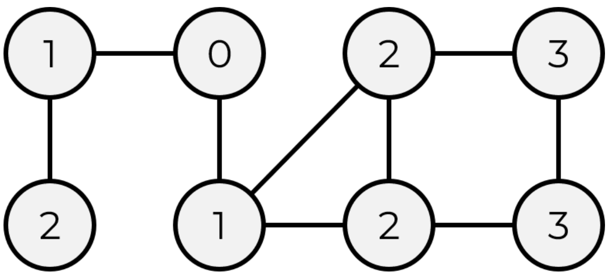
$$F = \{s, r, w\}$$



$$F = \{\cancel{s}, r, w\}$$



$$F = \{\cancel{r}, w\}$$
$$F = \{\cancel{w}, v\}$$
$$F = \{\cancel{v}, t, y\}$$



$$F = \{\cancel{t}, y\}$$
$$F = \{\cancel{y}, u\}$$
$$F = \{\cancel{u}, \cancel{z}\}$$

Exercice 4

Sujet

Soit G un graphe orienté sur lequel on effectue un parcours en profondeur d'abord depuis un sommet donné. Ce parcours ne permet pas d'atteindre tous les sommets du graphe : que pouvez-vous en déduire ?

Résolution

G n'est donc pas fortement connexe car il existe un sommet s qui ne peut pas atteindre tous les autres sommets.

Exercice 6

Sujet

Proposer une version itérative du parcours en profondeur.

Résolution

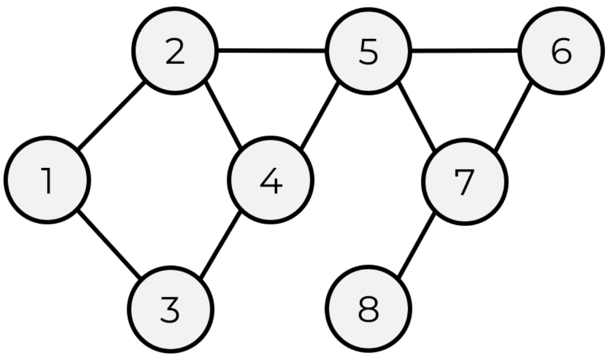
Nous pouvons remplacer les appels récursifs par l'utilisation d'une pile :

```
procédure pep_iteratif(Entrées G, s ; Sorties d, f, q)

début
    pour x ∈ S faire
        Couleur[x] ← blanc
        p[x] ← NUL;
    fin pour

    Couleur[s] ← gris;
    P ← ∅;
    empiler(P, s);
    d[s] ← 1;
    temp ← 2;

    tant que P ≠ ∅ faire
        x ← sommet(P);
        si x a un voisin y non traité alors
            empiler(P, y);
```

```
            Couleur[y] ← gris;
            d[y] ← temp;
            temp ← temp + 1;
            p[y] ← x;
        sinon
            Couleur[x] ← noir;
            f[x] ← temp;
            temp ← temp + 1;
            depiler(P);
        fin si
    fin tant que
fin
```

Exercice 7

On considère le graphe non orienté simple $G = (S, A)$ suivant :



1. Est-ce que les listes $L_1 = (2,5,4,3,1,6,7,8)$ et $L_2 = (5,7,8,2,4,1,3,6)$ qui donnent l'ordre dans lequel les sommets ont été découverts, peuvent être obtenues par l'algorithme de parcours en profondeur ?
2. Même question pour le parcours en largeur, avec les listes $L_1 = (6,5,7,8,2,4,1,3)$ et $L_2 = (1,2,3,4,5,6,7,8)$.

Résolution

Question 1

- L_1 : oui, car on a bien découvert les sommets dans l'ordre donné par la liste.
- L_2 : non, car après 8 on revient à 7 et on a découvert 6 et non 2.

Question 2

- L_1 : non, car on empile 5 avant 7 donc on doit traiter 2 avant 8.
- L_2 : oui, car on a bien découvert les sommets dans l'ordre donné par la liste.