THEORIE DES GRAPHES

5) PARCOURS DES GRAPHES

Il existe deux stratégies d'exploration d'un graphe, le parcours en profondeur et le parcours en largeur.

5.1. Parcours en profondeur d'un graphe orienté

Soit G un graphe orienté dont tous les sommets sont initialement non marqués. Le principe du parcours en profondeur est le suivant. On choisit un sommet de départ s, on le marque et on suit un chemin issu de s, aussi loin que possible, en marquant les sommets au fur et à mesure qu'on les rencontre pour la première fois. Arrivé en fin de chemin, on revient au dernier choix fait, et on prend une autre direction. Enfin, on réitère le processus et ce, tant qu'il existe des sommets de G non marqués. Le schéma de l'algorithme est le suivant.

```
Procédure parcours-prof (d G : GRAPHE) ;
{...}
Var i : entier ;
marque : tableau [1 ... n] de booléens ;
Début
        pour i = 1 à n faire
                marque [i] := faux ;
        pour i = 1 à n faire
                Si marque[i] = faux alors
                        prof (i, G, marque)
 Fin ;
```

où prof(i, G, marque) est une procédure récursive de parcours en profondeur qui rencontre tous les sommets de G accessibles depuis i et les marque lors de la première rencontre.

```
Procédure prof (d i : entier ; d G : GRAPHE ; r M : tableau [1 ... n] de booléens) ;

{cette procédure rencontre tous les sommets de G accessibles depuis i et les
marque lors de la première rencontre. M est un tableau de marques}

Var j : entier ; {j et i sont des sommets}

Début

  M[i] := vrai ; {1ère rencontre de i, on le marque} {écriture en ordre préfixe}

  Pour j := 1 à n faire {tous les j : G[i, j] = vrai, c.-à-d., tous les successeurs j de i}

        Si (G[i, j] = vrai) et (M[j] = faux) alors

                    prof (j, G, M) ;

  Fin

  {dernière rencontre de i} {écriture en ordre suffixe}

Fin ;
```

Remarque

Dans le schéma de l'algorithme on a supposé que G est représenté par sa matrice d'adjacence G[i, j]. Si maintenant, on utilise la représentation par listes de successeurs, c.-à-d.,

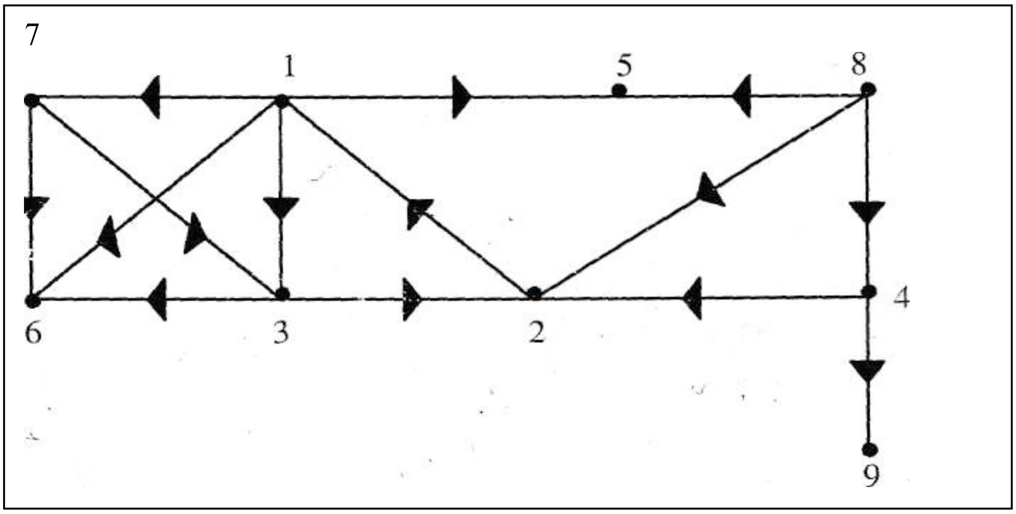
```
type adr = ^doublet

doublet = enregistrement

    no : entier

    suiv : adr

Fin ;

GRAPHE = tableau [1 … n] d’adr
```

Procédure prof (d i : entier ; d G : GRAPHE ; r M : tableau [1 ... n] de booléens) ;

{cette procédure rencontre tous les sommets de G accessibles depuis i et les marque lors de la première rencontre. M est un tableau de marques}

Exemple

Considérons le graphe G donné par la figure 1, et supposons que l'ordre d'exploration des successeurs d'un sommet de G soit l'ordre des numéros croissants.



<u>Figure 1. Un exemple d'un graphe G</u>.

On commence par l'appel de prof(l, G, M) qui marque le sommet 1 et choisit 3, le premier successeur de 1. Comme 3 est non marqué, il y a appel récursif de prof(3, G, M). Le sommet 3 est alors marqué, et son premier successeur, à savoir, 2 est choisi. Comme 2 est non marqué, il y a appel récursif de prof(2, G, M). Le sommet 2 est alors marqué, et son premier successeur 1 est choisi. Comme 1 est déjà marqué, et que 1 est le seul successeur de 2, la procédure prof(2, G, M) s'arrête, et la visite des sommets reprend au deuxième successeur de 3, soit au sommet 6. Comme 6 n'admet pas de successeur, prof(6, G, M) marque 6 et s'arrête. Tous les successeurs de 3 ont été explorés, donc prof(3, G, M) se termine, et prof(l, G, M) se poursuit par le choix de 5 qui n'est pas encore marqué. prof(5, G, M) marque 5 et se termine de suite, car 5 n'admet pas de successeur. Comme le successeur suivant de 1, à savoir, 6 est déjà marqué, prof(1, G, M) choisit 7, le dernier successeur de 1 non marqué. prof(7, G, M) marque 7 et se termine, car tous les successeurs de 7 ont déjà été marqués. La procédure prof(l, G, M) est maintenant terminée et donc, on retourne à la procédure parcours-prof(G) qui conduit à l'appel de prof(4, G, M), car 4 est le plus petit sommet non encore marqué. Comme le premier successeur de 4, à savoir, 2 est déjà marqué, prof(4, G, M) marque 4 et appel prof(9, G, M) qui marque 9. Comme 9 n'admet pas de successeur, prof(9, G, M) se termine. La procédure prof(4, G, M) est maintenant terminée et donc, on retourne à parcours-prof (G) qui conduit à l'appel de prof(8, G, M). prof(8, G, M) marque 8 et se termine, car tous les successeurs de 8 ont déjà été marqués. Comme tous les sommets de G sont marqués, l'algorithme s'achève.

Pour l'algorithme de parcours en profondeur, l'opération fondamentale est l'examen des marques. L'instruction à considérer pour déterminer le nombre total d'examens des marques est

"Pour i = 1 à n faire Si marque[i] = faux alors prof(i, G, marque)". Donc, pour la procédure parcours-prof, le nombre total d'examens des marques est égal au nombre d'examens des marques faits lors des n appels de la procédure prof, plus les n examens de marques qui correspondent à "marque[i]=faux"

Or, comme l'examen des marques dans la procédure prof(i, G, marque) se fait pour chaque successeur j de i, et comme, par ailleurs, on sait qu'avec une représentation par matrice d'adjacence, la complexité en temps de la recherche des successeurs d'un sommet i est en $\Theta(n)$, on conclut qu'avec une représentation par matrice d'adjacence, la complexité en temps de la procédure parcours-prof est en $\Theta(n + n^2)$, soit en $\Theta(n^2)$.

Notons que pour un graphe non orienté, l'algorithme de parcours en profondeur précédent reste valable et sa complexité reste également en en $\Theta(n^2)$.

5.2. Forêt couvrante associée au parcours en profondeur d'un graphe orienté

On appelle **arcs couvrants** les arcs x → y dans G tels que prof(x, G, M) appelle prof(y, G, M). Au parcours en profondeur d'un graphe, on associe une forêt couvrante d'arborescences, c.-à-d., une suite ordonnée d'arborescences, chacune composée d'arcs couvrants.

Exemple

La forêt couvrante associée au parcours en profondeur du graphe G de la figure 1 est celle qui suit.

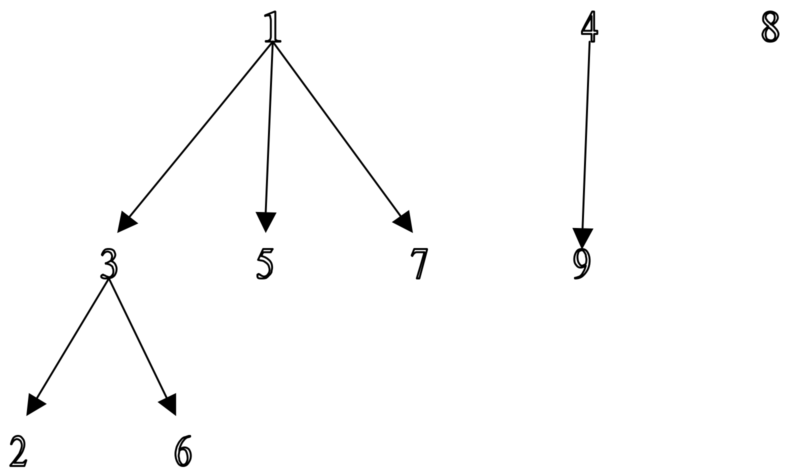


Figure 2. Forêt couvrante associée au parcours en profondeur du graphe G de la figure 1.

Notons que si l'on insère dans la procédure prof une instruction d'écriture de s, au moment de la première rencontre, on obtient alors l'impression des sommets dans l'ordre d'un parcours préfixe de la forêt couvrante. Par exemple, l'ordre préfixe sur la forêt couvrante de la figure 2 est 1 3 2 6 5 7 4 9 8.

Par contre, si cette instruction d'écriture est insérée au moment du dernière rencontre, on obtient alors l'impression des sommets dans l'ordre d'un parcours suffixe (ou postfixe) de la forêt couvrante. Par exemple, l'ordre suffixe sur la forêt couvrante de la figure 2 est 2 6 3 5 7 1 9 4 8.

Lors du parcours en profondeur d'un graphe orienté G, on distingue quatre types d'arcs : les arcs couvrants, les arcs en arrière, les arcs en avant et les arcs croisés.

On appelle **arcs en arrière** les arcs dont l'extrémité terminale est un ascendant de l'extrémité initiale dans la forêt couvrante. Autrement dit, ce sont des arcs x→y dans G tels que, dans la forêt couvrante, y est un père de x (y = x est possible). Par exemple, pour la figure 2, l'arc 2→1 est un arc en arrière car 1 est un père de 2 dans la forêt couvrante.

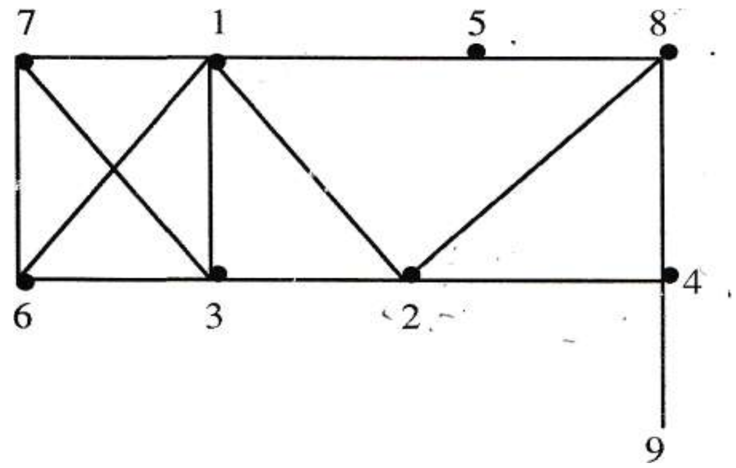
On appelle **arcs en avant** les arcs x→y dans G tels que, dans la forêt couvrante, y est un descendant propre de x, mais pas un fils direct de x. Par exemple, pour la figure 2, l'arc 1→6 est un arc en avant.

On appelle **arcs croisés** les arcs x→y dans G tels que, y n'est ni un père, ni un fils de x dans la forêt couvrante. Autrement dit, ce sont des arcs pour lesquels il n 'existe pas de chemin entre leurs extrémités dans la forêt couvrante. Par exemple, pour la figure 2, les arcs croisés sont 4→2, 7→6, 7→3, 8→5, 8→2 et 8→4.

Notons que pour un graphe non orienté, on parle encore d'arcs pour la forêt couvrante. Cela tient au sens du parcours qui implique une orientation des arêtes.

Exemple

Considérons le graphe non orienté G correspondant au graphe de la figure 1.



Pour ce graphe G, la forêt couvrante associée est celle qui suit.

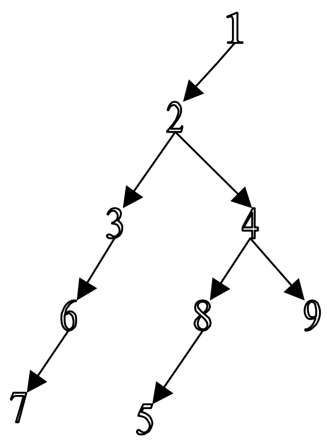


Figure 3. Un exemple de forêt couvrante associée à un graphe non orienté.

Parcours en largeur d'un graphe orienté

Soit G un graphe dont tous les sommets sont initialement non marqués. Appelons distance de s à x la longueur d'un plus court chemin de s à x. Le principe du parcours en largeur est le suivant. On choisit un sommet de départ s et on le marque. Ensuite, on marque tous les sommets qui se trouvent à une distance 1 de s, puis tous ceux qui se trouvent à une distance 2 de s et ainsi de suite. Enfin, on réitère le processus et ce, jusqu'à ce que tous les sommets de G soient marqués.

Notons que comme pour le parcours en profondeur, on peut associer au parcours en largeur une forêt couvrante.

Exemple

La forêt couvrante associée au parcours en largeur du graphe G de la figure 1 est celle qui suit.

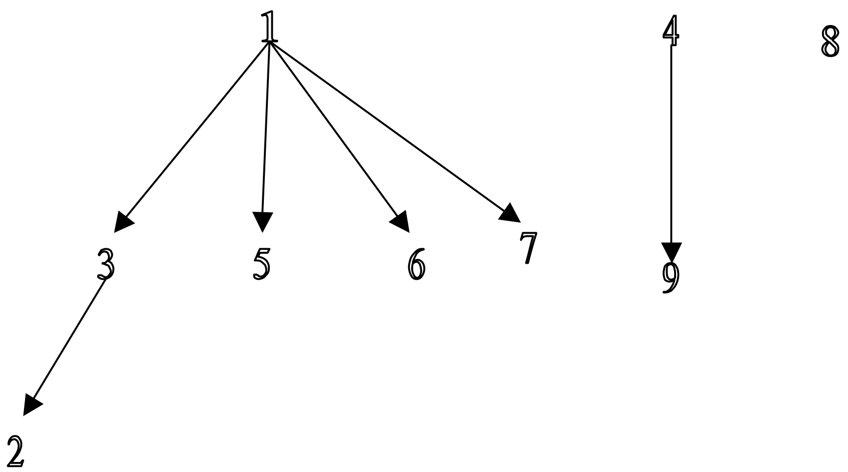


Figure 4. Forêt couvrante associée au parcours en largeur du graphe G de la figure 1.