Parcours en profondeur

Alix Munier-Kordon et Maryse Pelletier

LIP6
Sorbonne Université
Paris

LU2IN003 Initiation à l'algorithmique

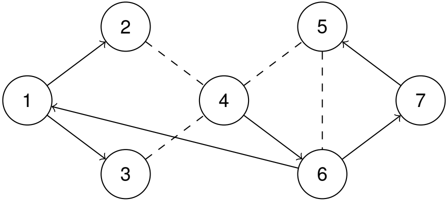
Plan du cours

1. Parcours en profondeur d'un graphe non orienté

2. Parcours en profondeur d'un graphe orienté

Principe général du parcours en profondeur

Soit $G = (V, E)$ un graphe non orienté et un sommet $s \in V$. A partir de l'origine s, on visite les sommets le long d'une chaîne élémentaire jusqu'à arriver à un sommet sans voisin non visité. On revient alors au dernier sommet sur la chaîne avec un voisin non visité et on recommence.
(ce n'est pas une définition)



$L = (4, 6, 7, 5, 1, 2, 3)$ est un parcours en profondeur d'origine 4.

Parcours en profondeur

Definition (Parcours en profondeur)

Soit $G = (V, E)$ un graphe non orienté connexe et $L = (v_1, \ldots, v_n)$ un parcours de G d’origine v_1.
L est un parcours en profondeur si pour tout sous-parcours $L_k = (v_1, \ldots, v_k)$ avec $k < n$, v_{k+1} est un sommet adjacent du **dernier** sommet ouvert de L_k.

Pour le parcours en profondeur $L = (4, 6, 7, 5, 1, 2, 3)$ du graphe précédent,

- Pour $L_4 = (4, 6, 7, 5)$, 5 et 7 sont fermés. Le dernier sommet ouvert est 6 et 1 est un adjacent de 6.
- Pour $L_5 = (4, 6, 7)$, 7 est ouvert et 5 est un sommet adjacent à 5.

Graphe de liaison en profondeur

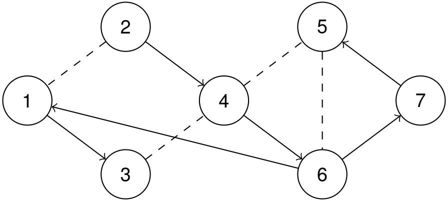
Definition (Graphe de liaison en profondeur)

Soit $G = (V, E)$ un graphe non orienté connexe et $L = (v_1, \ldots, v_n)$ un parcours en profondeur de G d'origine v_1. Le graphe orienté $\mathcal{A}^\star(L) = (V(L), H(L))$ est le graphe de liaison en profondeur de L si :

- $\mathcal{A}^\star(L) = (V(L), H(L))$ est un graphe de liaison de L;
- Pour tout sous-parcours $L_k = (v_1, \ldots, v_k)$ avec $k < n$, v_{k+1} a pour prédécesseur le sommet ouvert plus grand indice de L_k.

Pour tout parcours en profondeur, le graphe de liaison est unique.

Graphe de liaison en profondeur



Est-ce que le graphe de liaison (représenté par les flèches) correspond à un parcours en profondeur ? Est-il unique ?

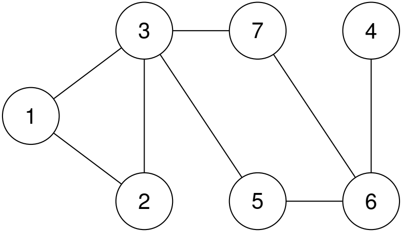
Algorithme récursif de construction d'un parcours en profondeur

Require: Un graphe non orienté $G = (V, E)$, un sommet s
Ensure: Un parcours en profondeur L d'origine s

```
function DFS(G, s)
visite[s] := True, L := (s)
for all arete {s, u} ∈ E do
    if not visite[u] then
        L := L+DFS(G, u)
    end if
end for
return L
end function
```

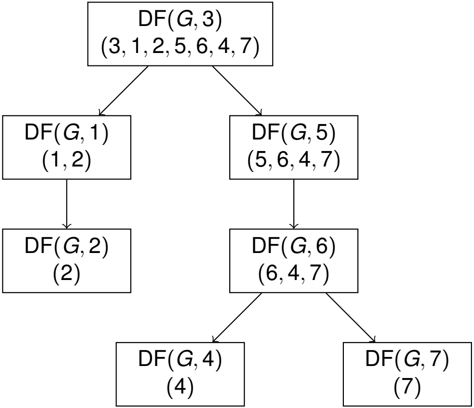
visite est un tableau de booléens sur les sommets dont les cases sont initialisées à *Faux*.

Exemple d'exécution de l'algorithme de construction d'un parcours en profondeur



Exécutez DFS(G, 3). Pour cela, donnez l'arbre des exécutions, et le parcours en profondeur obtenu. Le cas échéant, prendre en priorité le sommet de plus petit numéro.

Exécution de DFS(G, 3) avec valeur de retour



u	sommets visités
3	$\{3\}$
1	$\{1, 3\}$
2	$\{1, 2, 3\}$
5	$\{1, 2, 3, 5\}$
6	$\{1, 2, 3, 5, 6\}$
4	$\{1, 2, 3, 4, 5, 6\}$
7	$\{1, 2, 3, 4, 5, 6, 7\}$

Arbre des appels et évolution de l'ensemble des sommets visités.

Instants de pré-visite, post-visite

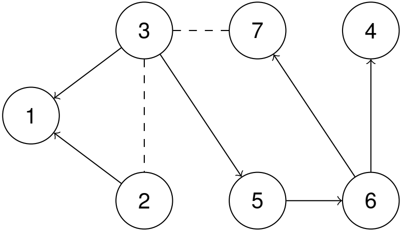
Definition (Instants de pré-visite, post-visite)

Soit $G = (V, E)$ un graphe non orienté, L un parcours en profondeur et $n = |V|$. On numérote de 1 à $2n$ tous les instants où l'on rentre et l'on sort d'un appel de DFS.
Les instants de pre-visite et post-visite sont alors deux fonctions *pre* et *post* de $V \to \{1, \ldots, 2n\}$ telles que pour tout $v \in V$:

- $pre(v)$ est l'instant où le sommet v est visité (*i.e* instant où on appelle DFS(G,v);
- $post(v)$ est l'instant où on sort de DFS(G,v).

On représente en générale ces deux valeurs par un intervalle $[pre(v), post(v)]$.

Arbre associé à un parcours et instants de pré-visite et post-visite



u	$[pre(v), post(v)]$
3	$[1, 14]$
1	$[2, 5]$
2	$[3, 4]$
5	$[6, 13]$
6	$[7, 12]$
4	$[8, 9]$
7	$[10, 11]$

Arbre des exécutions et instants de pré-visite et post-visite pour le parcours en profondeur $L = (3, 1, 2, 5, 6, 4, 7)$

Propriété fondamentale des intervalles $[pre, post]$

Theorem

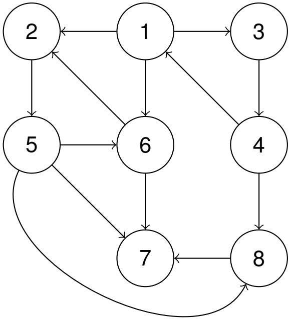
Pour tout couple de sommets $(u, v) \in V^2, u \neq v$, les intervalles $[pre(u), post(u)]$ et $[pre(v), post(v)]$ sont tels que :

- $[pre(u), post(u)] \cap [pre(v), post(v)] = \emptyset$, *ou*
- $[pre(u), post(u)] \subset [pre(v), post(v)]$, *ou*
- $[pre(v), post(v)] \subset [pre(u), post(u)]$.

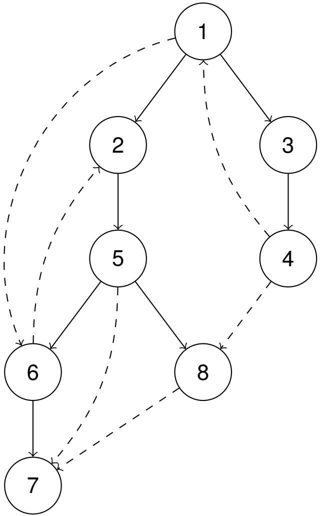
Par exemple,

- Pour $u = 2$ et $v = 6$, $[3, 4] \cap [7, 12] = \emptyset$;
- Pour $u = 5$ et $v = 7$, $[10, 11] \subseteq [7, 13]$.

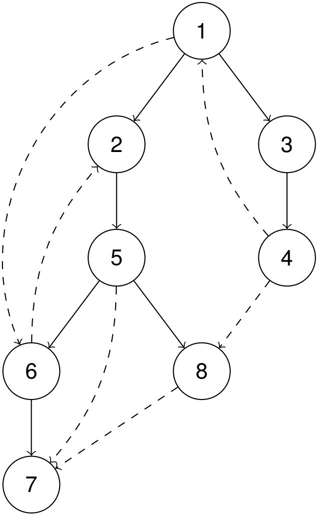
Passage à un graphe orienté



$L = (1, 2, 5, 6, 7, 8, 3, 4)$ est un parcours en profondeur d'origine 1.



Passage à un graphe orienté



u	$[pre(v), post(v)]$
1	$[1, 16]$
2	$[2, 11]$
5	$[3, 10]$
6	$[4, 7]$
7	$[5, 6]$
8	$[8, 9]$
3	$[12, 15]$
4	$[13, 14]$

$$L = (1, 2, 5, 6, 7, 8, 3, 4)$$

Parcours en profondeur d'un graphe orienté

Definition (Parcours en profondeur d'un graphe orienté)

Soit $G = (V, E)$ un graphe orienté et $L = (v_1, \ldots, v_n)$ un parcours de G d'origine v_1.
L est un parcours en profondeur si pour tout sous-parcours $L_k = (v_1, \ldots, v_k)$ avec $k < n$, v_{k+1} est un sommet successeur du **dernier** sommet ouvert de L_k.

- 8 est un successeur de 5, qui est le dernier sommet ouvert de $L = (1, 2, 5, 6, 7)$;
- 3 est un successeur de 1 qui est le dernier sommet ouvert de $L = (1, 2, 5, 6, 7, 8)$

Descendant, Ancêtre

Soit $\mathcal{A} = (V, A)$ une arborescence.

Definition (Descendant)

Soient deux sommets $(u, v) \in V^2$ avec $u \neq v$. u est un descendant de v si il existe un chemin dans $\mathcal{A}$ de v à u.

Definition (Ancêtre)

Soient deux sommets $(u, v) \in V^2$ avec $u \neq v$. u est un ancêtre de v si il existe un chemin de u à v.

Partition des arcs en fonction d'un parcours en profondeur

Soit $G = (V, A)$ un graphe orienté et L un parcours en profondeur de G. Les arcs du graphe sont partitionnés en 4 ensembles :

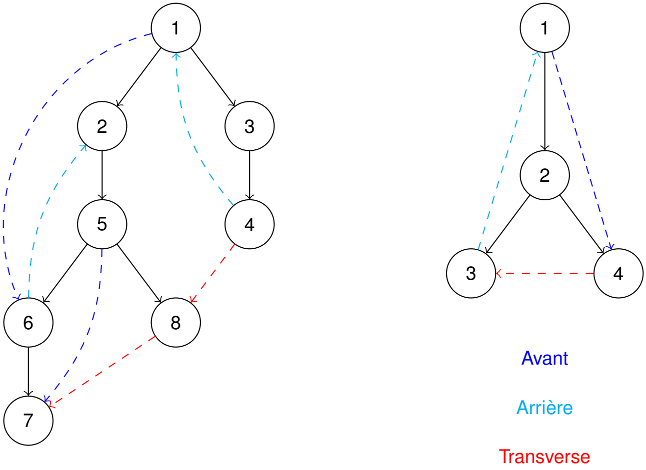
arcs de liaison : l'ensemble $H(L) \subset A$ des arcs du graphe de liaison du parcours en profondeur $\mathcal{A}^\star(L)$;

arcs avant : l'ensemble $F(L)$ des arcs $(u, v) \subset A - H(L)$ tels que v est un descendant de u dans $\mathcal{A}^\star(L)$;

arcs arrières : l'ensemble $I(L)$ des arcs $(u, v) \subset A - H(L)$ tels que v est un ancêtre de u dans $\mathcal{A}^\star(L)$;

arcs transverses : l'ensemble $C(L) = A - (H(L) \cup F(L) \cup I(L))$.

Partition des arcs en fonction d'un parcours en profondeur



$$L = (1, 2, 5, 6, 7, 8, 3, 4)$$

Détection des circuits dans un graphe orienté

Theorem

Soit $G = (V, A)$ un graphe orienté et L un parcours en profondeur de G. Soit G possède un circuit si et seulement si le parcours L possède au moins un arc arrière.

Ce théorème permet donc d'écrire un algorithme qui détecte la présence de circuit dans un graphe.

Conclusion sur le parcours en profondeur

- Algorithme fondamental pour les graphes orientés ou non;
- Permet de détecter la présence de circuit dans un graphe orienté;
- A la base de nombreuses applications non vues dans ce cours (construction d'un tri topologique, recherche des composantes fortement connexes).