Mathématiques disctères 2 - graphes

Cours 2 - Parcours de graphes

N. de Rugy-Altherre

1. Généralités
2. Parcours en largeur (BFS)
3. Parcours en profondeur (DFS)

Le but d'un parcours de graphe

- Visite de tous les sommets accessibles depuis un sommet de départ donné
- Exemple de problème : comptez le nombre de sommets de la composante connexe ayant une certaine propriété.
- Les algorithmes de parcours sont *locaux* : un graphe est vu comme composé d'un sommet et des graphes enracinés dans ses voisins.

Gagnerait-on quelque chose à avoir un accès global au graphe ?

- Exemple de problèmes : comptez le nombre de sommet de la composante connexe.
- Modèle du graphe : l'ensemble V des sommets et une fonction $E(u, v)$ déterminant si (u, v) est une arête.
- Un algorithme :

```
Entrees : V, E, v0
Variable :
        visiste : ensemble de sommets
Debut
        Pour w dans V faire
                Si E(v,w) et non(w dans visite) Alors
                                visite.ajouter(w)
```

et... Heu... On ne gagne rien...

Le fonctionnement d'un parcours de graphe

1. Comment parcourir un graphe ?
   - Marquage des sommets par des couleurs :
     - Blanc = sommets pas encore visités
     - Gris = sommet en cours de traitement
     - Noir = sommet qu'on a traité
   - Au début, le sommet de départ est gris, les autres blancs.
   - À chaque étape, un sommet gris est sélectionné
     - Si tous ses voisins sont déjà gris ou noirs, alors il est colorié en noir
     - Sinon on colorie un (ou plusieurs) de ses voisins blancs en gris

   Jusqu'à ce que tous les sommets soient noirs ou blancs.
2. Mise en oeuvre : stockage des sommets gris dans une structure

   - Si on utilise une file (FIFO), alors c'est un parcours en largeur
   - Si on utilise une pile (LIFO), alors c'est un parcours en profondeur

Vocabulaire

- Parcours en profondeur : depth-first search (DFS)
- Parcours en largeur : breadth-first search (BFS)

Spécification d'un algorithme de parcours

1. Fonction `parcours`
   - Entrées : un graphe G et un sommet v_0
   - Postcondition : l'arborescence du parcours de G à partir du sommet v_0
2. arborescence associée à un parcours :
   - v est le père de w dans l'arbre de parcours si c'est v qui a colorié w en gris.
   - v est la racine $v = v_0$ ou s'il n'existe pas de chemin entre v et v_0
   - Mémorisation de l'arborescence par exemple dans un tableau T tel que $T[v] = null$ si v est une racine, $T[v] = w$ si w est le père de v.

Exemple :

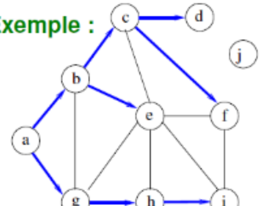

Tableau π correspondant :

-	a	b	c	b	c	a	g	h	-
a	b	c	d	e	f	g	h	i	j

1. Généralités
2. Parcours en largeur (BFS)
3. Parcours en profondeur (DFS)

Étude de BFS : complexité

Complexité

$\mathcal{O}(n+p)$ pour un graphe ayant n sommets et p arcs, sous réserve d'une implémentation par liste d'adjacence.

Cette étude repose sur la propriété suivante : chaque sommet et chaque arête n'est visitée par l'algorithme qu'une fois. Ce qui se démontre avec invariant de boucles de la ligne 10 :

1. Un sommet noir reste noir.
2. Un sommet gris reste gris ou devient noir.

Corolaire : chaque sommet est visité au plus une fois et chaque arête au plus deux fois.

Étude de BFS : arborescence

Soit $G = (V, E)$ un graphe (orienté ou non) qu'on parcourt par profondeur à partir de v_0

Définition

Le graphe de parcours est un graphe orienté $G' = (V', E')$ tel que

- $V' = \{v \in V, v$ et v_0 sont dans la même comp connexe$\}$
- $(v, w) \in E'$ si v est le sommet à partir duquel w a été colorié en gris.

Théorème

Le graphe de parcours d'un parcours en largeur est un graphe.

Lemme (vu en MD1) : un graphe $G = (V, E)$ est un arbre ssi tout couple de sommets $(u, v) \in V^2$ est relié par un unique chemin.

Étude de BFS : arborescence

Théorème

Le graphe de parcours d'un parcours en largeur est un graphe.

Démonstration (de la propriété) par l'absurde : suppose qu'il existe deux sommets v et w et deux chemins distincts allant de v à w. Deux choix :

1. Soit w a deux prédécesseurs dans le sous-graphe de parcours, ce qui est impossible (cf la démonstration de la complexité)
2. Soit w n'a qu'un prédécesseur z. Auquel cas il existe deux chemins distincts de v à z de taille strictement plus petit que les deux chemins distincts de v à w. On pose $w = z$ et on reprend le raisonnement et on se ramènera au point 1.

Étude de BFS : Plus court chemin

Soit $G = (V, E)$ un graphe (orienté ou non) parcourus en largeur à partir de v_0.

Définition et notation

- Soit $v \in V$. On notera $v.d$ la date où v est devenu gris (0 pour v_0, 1 pour ses successeurs, etc.)
- Notons $\delta(v, w)$ la distance du plus court chemin entre v et w.

Propriété

$\forall v \in V, \delta(v_0, v) = v.d$

Étude de BFS : Plus court chemin

Propriété

$\forall v \in V,\ \delta(v_0, v) = v.d$

Démonstration :

1. Pour tout graphe $G = (V, E)$, $\forall (v, w) \in E, \forall s \in V\ \delta(s, w) \leq \delta(s, v) + 1$ (démonstration directe)
2. Pour un graphe $G = (V, E)$ parcours en largeur à partir de v_0, $\forall v \in V,\ v.d \geq \delta(v_0, v)$ (par récurrence).
3. On suppose que lors de l'exécution du parcours, la file f soit composée des sommets $f =< v_1, \ldots, v_k >$ (v_1 la tête de la file, v_k sa queue). Alors $v_k.d \leq v_1.d + 1$ et $\forall i \in [1, k-1],\ v_i.d \leq v_{i+1}.d$ (démonstration par récurrence).

Étude de BFS : Plus court chemin

Propriété

$\forall v \in V,\ \delta(v_0, v) = v.d$

Démonstration : par l'absurde. Supposons qu'il existe $v \in V$ vérifiant la propriété P : $\delta(v_0, v) \neq v.d$. On peut supposer que v est un sommet pour lequel $\delta(v_0, v)$ est minimal parmis les sommets ayant cette propriété.

- Visiblement, $v \neq v_0$. D'après le lemme 2, $v.d \geq \delta(v_0, v)$.
- Soit u le sommet qui précéde immédiatement v dans le plus court chemin de v_0 à v. Donc $\delta(v_0, v) = \delta(v_0, u) + 1$.
- Comme v est le plus petit sommet vérifiant P, au sens de la distance à v_0, u ne vérifie par P et donc $u.d = \delta(v_0, u)$.
- En combinant ces deux propriétés on obtient :

$$v.d > \delta(v_0, v) = \delta(v_0, u) + 1 = u.d + 1 \qquad (1)$$

Étude de BFS : Plus court chemin

Propriété

$\forall v \in V,\ \delta(v_0, v) = v.d$

Démonstration : par l'absurde.
(suite)

- En combinant ces deux propriétés on obtient :

$$v.d > u.d + 1 \qquad (1)$$

- Plaçons nous au moment où u est dépilé (passe du gris au noir). Alors :
  1. Soit v est blanc et donc il est colorié à cet instant en gris. Donc $v.d = u.d + 1$, ce qui est en contradiction avec l'inégalité (1)
  2. Soit v est noir et d'après le lemme 3, $v.d \leq u.d$ ce qui est en contradiction avec l'inégalité (1)
  3. Soit v est gris et donc est avant dans la file f et donc d'après le lemme 3, $v.d \leq u.d$ ce qui est en contradiction avec l'inégalité (1)

Étude de DFS : complexité

Complexité

$\mathcal{O}(n+p)$ pour un graphe ayant n sommets et p arcs, sous réserve d'une implémentation par liste d'adjacence.

Démonstration : Laissé au lecteur.

Étude de DFS : Propriétés

Soit $G = (V, E)$ un graphe (orienté ou non) parcours en profondeur à partir de v_0.

Définitions

- Soit $v \in V$. Notons $v.d$ l'instant où v est devenu gris, l'instant *de découverte*.
- Soit $v \in V$. Notons $v.f$ l'instant où v est devenu noir, l'instant de *fin du traitement*.

- Propriété : $\forall v \in V,\ v.d < v.f$
- Propriété : $v_0.d = 0$
- Propriété : $\forall v \in V, 0 \leq v.d \leq 2|V|$ et $\forall v, w \in V, v \neq w \Rightarrow v.d \neq w.d$
- Propriété : $\forall v \in V, 0 \leq v.f \leq 2|V|$ et $\forall v, w \in V, v \neq w \Rightarrow v.f \neq w.f$
- Propriété : le graphe de parcours est une forêt orientée (pas nécessairement un arbre). *Démonstration laissée au lecteur.*

Étude de DFS : Classification des arêtes

Soit $G = (V, E)$ un graphe (orienté ou non) parcours en profondeur à partir de v_0.

Définitions

Les arêtes du graphe peuvent être classifiée en

- Les arcs de liaison sont les arêtes de la forêt de parcours.
- Les arcs arrières (u, v) sont les arêtes de G telles que u soit un ancêtre de v dans la forêt de parcours.
- Les arcs avant (u, v) sont les arêtes de G telles que v soit un ancêtre de u dans la forêt de parcours et qui ne sont pas des arcs de liaison.
- Les arcs transverses : ce sont les autres.

Étude de DFS : Classification des arêtes

Théorème

Dans un parcours en profondeur d'un graphe non orienté G, chaque arête est soit un arc de liaison soit un arc arrière.

Démonstration :

- Soit (u, v) une arête quelconque. On peut supposer que $u.d < v.d$.
- Alors v doit être découvert et son traitement terminé pendant que u est grise.
- Si l'arête (u, v) est d'abord explorée dans le sens u vers v, alors v est encore blanc (car sinon u sera noir). Donc (u, v) devient un arc de liaison.
- Si l'arête (u, v) est explorée dans l'autre sens, de v vers u, alors elle devient un arc arrière.

Référence

Algorithique de Cormen, Leiserson, Rivest et Stein. Dunod 2010