Université de Versailles – Saint-Quentin-en-Yvelines 2019-2020
Master 1 Informatique

Algorithmique de Graphes – Connexité et les parcours. Le TD du 13 Novembre 2019

Tatiana Babicheva. Link : https ://www.babichev.org/TD/

Graphe orienté

Définition 1. Graphe Orienté

> $G = (V, A)$ où V est un ensemble fini de sommets et où $A \subseteq V \times V$ est un ensemble de paires *ordonnées* de sommets appelées *arcs*.

- On note (u, v) l'arc entre les sommets u et v de G.
- Pour un arc $(u, v) \in A$, v est un *successeur* de u et u est un *prédecesseur* de v.
- L'arc (u, v) est un *arc sortant* du sommet u et un *arc entrant* du sommet v.

Définition 2. Adjacence

> Deux arcs sont adjacents s'ils ont une extrémité commune. Deux sommets sont adjacents s'il existe un arc dont les extrémités sont ces deux sommets.

Définition 3. Degré entrant

> Soit $G = (V, A)$ un graphe orienté. Le degré entrant d'un sommet $u \in V$, noté $d^+(u)$ est le nombre d'arcs entrants dans le sommet u.

Définition 4. Degré sortant

> Soit $G = (V, A)$ un graphe orienté. Le degré sortant d'un sommet $u \in V$, noté $d^-(u)$ est le nombre d'arcs sortants du sommet u.

Définition 5. Chemin dans un graphe orienté

> Soit $G = (V, A)$ un graphe orienté. Un chemin dans G est une suite de sommets $v_0 v_1 \ldots v_n$ telle que chaque couple de sommets successifs (v_{i-1}, v_i), $i \in 1..n$ est un arc de A. La longueur du chemin est le nombre d'arcs du chemin.

Définition 6. Circuit dans un graphe orienté

> Un circuit dans un graphe orienté est un chemin tel que $v_0 = v_n$.

Définition 7. Forte Connexité

> Soit $G = (V, A)$ un graphe orienté. G est *fortement connexe*, si pour toute paire ordonnée de sommets distincts $u \in V$ et $v \in V$, il existe un chemin de u à v dans G.

Définition 8. Connexité au sens des arcs

> Soit $G = (V, A)$ un graphe orienté tel que $|V| > 1$. $F \subseteq A$ est un ensemble d'arcs déconnectant G si $(V, A \setminus F)$ n'est pas fortement connexe.
>
> $\lambda(G)$ est le cardinal du plus petit ensemble d'arcs déconnectant G.

Définition 9. **k-fortement connexe**

Soit $G = (V, A)$ un graphe orienté tel que $|V| > 1$. G est *k-fortement connexe* si $\lambda(G) \geq k$.

Exercice 1 *Forte connexité.*

1. Montrer que si un graphe orienté G est fortement connexe, alors chaque arc de G appartient à un circuit.
   From Tatiana

   On peut aller d'un coté de cet arc vers autre coté. Et alors on peut construire un cycle qui a cet arc.

2. Montrer qu'un graphe G est fortement connexe si et seulement si chaque sommet de G est la racine d'une arborescence et d'une anti-arborescence de G.
   From Tatiana

   arborescence — from
   anti-arborescence — to
   trop facile

3. Un graphe orienté G est dit *fortement cyclique* si pour chaque paire de sommets x, y de G, il y a une séquence de circuits $C_1, \ldots, C_k$ tels que x est dans C_1, y est dans C_k, et C_i et C_{i+1} ont au moins un sommet commun. Montrer que G est fortement connexe si et seulement si G est fortement cyclique. **From Tatiana**

   Si le graphe est fortement cyclique — on peut construire le chemin de chaque sommet vers chaque autre sommet.
   Si le graphe est fortement connexe — on peut construire les chemins de deux cotés pour chaque paire de sommets x et y. Si cette construction est un cycle qui ne s'intersecte pas, on a fini le preuve. Si on passe par les mêmes sommets — on peut produire les petites cycles.

4. Proposer un algorithme pour décider si un graphe est fortement connexe et un autre pour trouver un cycle. **From Tatiana**

   Par exemple, on peut utiliser algorithme de Tarjan.
   Tarjan

Exercice 2

Soit G un graphe orienté k-fortement connexe avec $k \geq 2$. Soit G' un graphe obtenu en ajoutant un nouveau sommet x à G ainsi que les arcs suivants :

— k arcs sortants de x vers k sommets distincts de G ;

— k arcs entrants en x à partir de k sommets distincts de G.

Montrer que G' est k-fortement connexe.
From Tatiana

Nous pouvons supprimer k arcs pour déconnecter le graphe : par exemple, k nouveaux arcs entrants

On ne peux pas supprimer moins : par example, on peut le prouver par l'absurde. Comme le graph initial est k-fortement connecte, le partie de nouveau graphe qui correspond à ancien va rester fortement connexe même après avoir supprimé les arcs (parce que on a supprimé en total moins de k arcs). De la nouvel partie, on supprime aussi moins de k arcs, alors il nous reste à minimum 1 arc sortant de nouvel vertex et à minimum 1 arc entrant à nouvel vertex. Comme ça, entre chaque pair des sommets nous pouvons toujours construire un chemin.

Exercice 3

Soit $n \geq 2$. Quel est le nombre maximum d'arêtes d'un graphe non orienté G à n sommets vérifiant $\lambda(G) = 1$?

From Tatiana

L'example maximale est K_{n-m}, K_m et pont : Comme ça, en totale il y a $\frac{(n-m-1)(n-m)}{2} + \frac{(m-1)(m)}{2} + 1$ arêtes, qui donne le problème de maximisation suivant.

$\max_{m \in (0,n)}(\frac{(n-m-1)(n-m)}{2} + \frac{(m-1)(m)}{2} + 1) =$

$= \max_{m \in (0,n)}(\frac{(n^2-2nm+2m^2-n}{2} + 1)$

Alors, $\frac{\delta(n^2-2nm+2m^2-n)}{\delta m} = 0$, donc $m = \frac{n}{2}$.

Exercice 4

Montrer que si un graphe non orienté G contient deux arbres couvrants arête-disjoints, alors $\lambda(G) \geq 2$. La réciproque est-elle vraie ?

From Tatiana

à un coté : par l'absurde

Non, parce que nous pouvons construire un cycle C_3.

Exercice 5

Étant donné deux entiers $1 < k < n$, on note $m(k,n)$ le nombre minimum d'arêtes d'un graphe non orienté k-connexe à n sommets.

1. Montrer que $m(k,n) \geq kn/2$.

   From Tatiana

   Comme le graphe est k-connexe, le degré de sommet est au minimum k

2. On suppose dans cette question que k et n sont pairs. On définit le graphe $G_{k,n}$ sur les sommets $\{0, \ldots, n-1\}$ par : $[i,j] \in E$ si et seulement si $i - j \in \{-k/2, \ldots, k/2\} \setminus \{0\}$ (l'addition est prise modulo n). Montrer que $G_{k,n}$ est k-connexe. En déduire la valeur de $m(k,n)$ pour k et n pairs (vérifiant $1 < k < n$).

   From Tatiana

   Si on supprime quelque k d'un des sommets alors ça va, $m(k,n) = kn/2$

3. Adapter la construction de la question précédente aux couples d'entiers quelconques vérifiant $1 < k < n$. En déduire la valeur de $m(k,n)$ pour tous ces couples d'entiers.

   From Tatiana

   si k est impair et n est pair on ajoute les diagonales

   si k est pair - le même

   si k et n sont impairs alors les diagonales plus une autre pour faire plus 1.

   $m(k,n)$ est le plus petit entier $\geq kn/2$

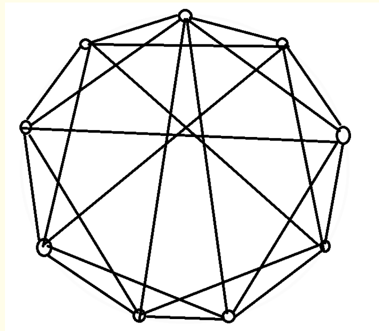
4. Montrer que $\lambda(G_{k,n}) = k$.

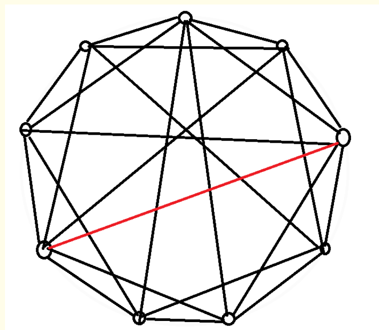
   From Tatiana

   Théorème de Menger

5. Dessiner deux graphes 5-connexe à 9 sommets non isomorphes.

   From Tatiana





Représentation

Matrice d'adjacence

- Les n sommets du graphe sont numérotés de 1 à n.
- Mémorisation dans une matrice (M) carrée $n \times n$ des arêtes (arcs) :
  - $M[i][j] = 1$ s'il existe une arête (un arc) entre les sommets i et j dans G
  - $M[i][j] = 0$ sinon.

Listes d'adjacence

- Les n sommets du graphe sont numérotés de 1 à n.
- Pour chaque sommet : mémorisation de la liste de ses voisins sous forme d'une liste chainée.

Parcours en Largeur

- Le graphe $G = (V, E)$ est représenté par des listes d'adjacence
- Pour chaque sommet u on a :
  - **Couleur[u]** = { blanc, gris, rouge }
  - **Parent[u]**
  - **Distance[u]** : distance depuis s dans le parcours.
- une file pour insérer les sommets découverts.

Parcours en largeur

- Chaque sommet est ajouté et supprimé au plus une fois dans la file.
- La liste d'adjacence de chaque sommet est balayée au plus une fois
- Le parcours en largeur s'éxécute en $O(|V| + |E|)$.

Parcours en profondeur

- Le graphe $G = (V, E)$ est représenté par des listes d'adjacence
- Z un ensemble.
- P une pile
- Chaque sommet est ajouté et supprimé au plus une fois dans la pile.
- La liste d'adjacence de chaque sommet est balayée au plus une fois
- Le parcours en profondeur s'éxécute en $O(|V| + |E|)$.

Exercice 6

Une entreprise d'informatique souhaite lancer la construction d'un nouvel ordinateur. La conception d'un tel ordinateur implique son passage successif sur différentes machines $\{m_1, m_2, \ldots, m_k\}$, avec un passage unique sur chaque machine.

Afin de planifier cette construction, on considère un graphe orienté G dans lequel :

- les sommets sont les machines,
- un arc (m, m') dans G signifie que l'ordinateur doit impérativement être traité par la machine m avant d'être traité par la machine m'.

1. Quelle propriété de G indiquerait que la construction est logiquement impossible ? Concevez un algorithme, utilisant un parcours en largeur, qui dit si cette propriété est ou non vérifiée pour G.

   From Tatiana

2. On considère à présent que la construction est possible. On désire indiquer un ordre de passage de l'ordinateur sur les machines. Pour ce faire, une machine m est numérotée i si elle est la i^{eme} machine utilisée. Quelle propriété doit vérifier une telle numérotation des sommets de G ? Concevez un algorithme, utilisant un parcours en profondeur, qui donne une numérotation possible des sommets de G. Pourquoi peut-on parler ici de "Tri topologique" ?

   From Tatiana