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

Algorithmique de Graphes – Chemin, distance, diamètre, connexité. Le TD du 6 Novembre 2019

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

Tous les graphes considérés dans cette section sont simples et non orientés. Les chemins et cycles considérés sont, sauf indication contraire, élémentaires : ils ne passent pas plusieurs fois par le même sommet.

Exercice 1 *Rayon, diamètre.*

On rappelle que l'excentricité d'un sommet u est défini par $e(u) = \max_{v \in V} d(u, v)$ où $d(u, v)$ est la distance de u à v (c'est-à-dire la longueur minimale d'un chemin de u à v). Le rayon du graphe G est défini par $r(G) = \min_{v \in V} e(v)$. Le diamètre de G est défini par $d(G) = \max_{u,v \in V} d(u, v)$.

1. Calculer le nombre maximum de sommets d'un graphe de degré au plus Δ et de rayon au plus k.
   From Tatiana

   On considère le centre. Le degré du centre est au plus Δ. Le rayon est au plus k, alors si on considère étoile du base le nombre de sommets de la génération 0 ne peux pas être supérieure de 1, génération 1 - de δ, génération 2 - de $delta \times (\delta + 1)$, etc.
   Pour un degré fixé il y a les articles intéressants (pour les avancés).
   Mooregraph
   https://core.ac.uk/download/pdf/81981341.pdf

2. Montrer que $r(G) \leq d(G) \leq 2r(G)$.
   From Tatiana

   La première inéquation est issue de la définition :
   $r(G) = \min_{v \in V} e(v) \leqslant \max_{u,v \in V} d(u, v) = d(G)$
   Pour prouver la deuxième, considérons $d(u, v) = d(G)$; $e(w) = r(G)$.
   En utilisant l'inégalité triangulaire, on obtient
   $d(G) = d(u, v) \leqslant d(u, w) + d(w, v) \leqslant e(w) + e(w) = 2r(G)$

3. Exhiber des graphes arbitrairement grands vérifiant $d(G) = r(G)$ et $d(G) = 2r(G)$.
   From Tatiana

   1) Tous les graphes complets (toutes les excentricités sont de 1)
   2) Le graphe 0-0-0-....-0. (chaîne simple avec longueur n)

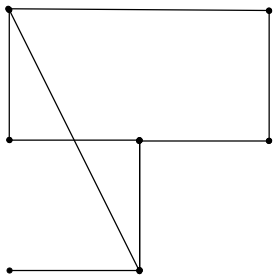
Exercice 2 *Centre*

On veut trouver un serveur dans un réseau qui soit le plus proche possible de tous les autres noeuds du réseau. Cette position s'appelle le centre dans un graphe (le sommet qui minimise le maximum des distances aux autres sommets). C'est le sommet d'excentricité minimale.

1. La distance entre deux sommets est le plus court chemin entre ces deux sommets. Proposer un algorithme qui trouve le plus court chemin d'un sommet à un autre. Il peut être plus facile de calculer simultanément le plus court chemin d'un sommet à tous les autres.
   From Tatiana

   Algorithme de Deikstra : dans les files supplémentaires.

2. Donner le centre et les deux sommets les plus distants du graphe suivant.



From Tatiana

one useful link (google translate helps)

3. Donner le centre d'une grille de taille n fois n (justifier).

From Tatiana

Nous introduisons un système de coordonnées cartésiennes. Si nous considérons le sommet $v = (x, y)$, alors $e(v) = \max(x, n - x) + \max(y, n - y)$ Alors on peut dire que $\max(a, b) \geqslant \frac{a+b}{2}$ Cela signifie que l'excentricité n'est pas inférieure à n pour tous les sommets et que le minimum est atteint si a et b sont égaux ; dans le cas de n impair, nous obtenons immédiatement la réponse. Dans le cas impair, nous voyons que lorsque nous nous éloignons du "centre", notre excentricité augmente, ce qui signifie que tous les centres sont situés à la même distance du "centre". Alors nous nous efforçons de la minimiser afin d'obtenir quatre points centrales

4. En utilisant l'algorithme de plus court chemin, donner un algorithme qui calcule le centre de n'importe quel graphe.

From Tatiana

```
const int  N = 100; // Nombre de sommets
const int   INF = 1E100; // Infinity
int         d[N][N]; // Distancions
int         e[N]; // excentricité de sommets
set <int>   c; // Centre de graphe
int         rad = INF; // rayon
int         diam; // diametr
```

From Tatiana

```
// Algorithme de Floyd-Warshall
for (int k = 0; k < N; k++) {
    for (int j = 0; j < N; j++) {
        for (int i = 0; i < N; i++) {
            d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
        }
    }
}
```

From Tatiana

```
// recherche de l'excentricité
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        e[i] = max(e[i], d[i][j]);
    }
}

// recherche du rayon et du diamètre
```

```
for (int i = 0; i < n; i++) {
    rad = min(rad, e[i]);
    diam = max(diam, e[i]);
}

for (int i = 0; i < n; i++) {
    if (e[i] == rad) {
        c.insert(i);
    }
}
```

Exercice 3 *Intersection des chemins de longueur maximale dans un graphe.*

1. Dans un graphe connexe, montrer que deux chemins de longueur maximale ont un sommet en commun.
   From Tatiana

   S'ils n'ont pas de sommet commun, alors, puisque le graphe est connecté, nous pouvons indiquer un sommet du premier et un sommet du second chemin, reliés par un chemin C de bords n'appartenant pas à nos chemins. Nommons les extrémités des chemins originaux avant et après les sommets sélectionnés par A1, A2 ; B1, B2. Au moins une des chemins A1-C-B1, A1-C-B2, A2-C-B1 et A2-C-B2 sera plus longue que l'originale. Contradiction.

2. Est-ce vrai dans un graphe non connexe ?
   From Tatiana

   Non, par exemple pour le graphe :
   0-0-0
   0-0-0

Exercice 4 *Tours de Hanoï.*

On dispose de trois piquets A,B,C, ainsi que de n disques de diamètres distincts. Une position est légale si tous les disques sont sur les piquets, et si aucun des disques ne repose sur un disque de diamètre inférieur. Un coup consiste à déplacer le disque se trouvant au sommet d'un piquet pour le placer au sommet d'un autre (la position obtenue doit être légale). On définit le graphe simple non orienté $G_n = (V_n, E_n)$ associé à ce jeu de la manière suivante :

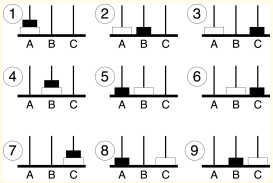
— V_n est l'ensemble des positions légales ;

— $[u, v] \in E_n$ si on peut passer de u à v en un coup.

Enfin, on définit deux positions particulières : la position *initiale* où tous les disques sont sur A, et la position *finale* où tous les disques sont sur B. On note d_n le nombre de coups nécessaires pour atteindre la position finale à partir de la position initiale ($d_n = +\infty$ si c'est impossible).

1. On commence par regarder le cas $n = 2$.
   — Dessiner G_2.
   From Tatiana

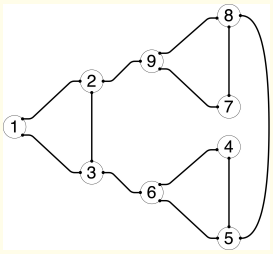
   Considérons toutes les positions légales :
   On va les dessiner et énumérer ici :



Les positions légales

From Tatiana

Alors, on peut obtenir le graphe :



— Calculer la distance de chaque position à la position initiale. Combien de coups sont nécessaires pour passer de la position initiale à la position finale dans G_2 ?

From Tatiana

Par Dijkstra on obtient qu'il faut passer 3 coups

— Choisir un sommet correspondant à une position où les deux disques ne sont pas sur le même piquet. Calculer la distance de cette position à chacune des autres.

From Tatiana

Par Dijkstra

— Par symétrie, en déduire le diamètre de G_2. Comparer à d_2.

From Tatiana

C'est 3. Alors, on obtient égalité.

On s'intéresse maintenant au cas général du jeu à n disques. Pour chacune des questions, on commencera par reformuler la question en utilisant la terminologie des graphes.

2. Calculer le nombre de positions légales.

From Tatiana

Par récurrence : on ajoute un disque — à une des trois positions, alors, 3^n

3. Quels sont les nombres minimaux et maximaux de coups possibles pour une position ?

From Tatiana

Il y a trois cas :

(a) Il y a des disques à chaque piquet. Les disques les plus hautes sont alors avec les diamètres 1, x et y, $x < y$ (l'ordre est inconnu). Alors, on peut déplacer le disque de diamètre 1 vers n'importe lequel des autres piquets (deux coups possibles) et le disque de diamètre x au-dessus du disque diamètre y (un coup possible). Au total il y a trois coups possibles.

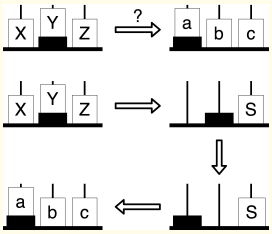
(b) Il y a des disques sur deux des trois piquets. Les disques les plus hautes sont alors avec les diamètres 1 et x (l'ordre est inconnue). Alors, on peut déplacer le disque de diamètre 1 vers n'importe lequel des autres piquets (deux coups possibles) et le disque de diamètre x vers le piquet vide (un coup possible). Au total il y a trois coups possibles.

(c) Toutes les disques sont sur un des trois piquets. Alors, on peut déplacer le disque de diamètre 1 vers n'importe lequel des autres piquets (deux coups possibles).

4. Montrer qu'une suite de coups permet de passer de n'importe quelle position à n'importe quelle autre.

From Tatiana

On peux le prouver par récurrence.
Considérons la base de la récurrence (induction) : Par exemple, c'est le graphe connexe G_2.
Disons que le graphe G_n est connexe.
Prouvons que le graphe $G_{(n+1)}$ est connexe.
Tous les positions dans le graphe G_n sont connectées.
Alors, on ajoute le disque avec un diamètre supérieur dans le système.
Ici je montre un des algorithmes possibles :
x,y,z,a,b et c sont les transitions légales des disques sur un piquet (ils peuvent être vides).
S est une séquence de 1 à n sur un piquet.



5. Combien de coups sont nécessaires pour passer de la position finale à la position initiale ? D'autres couples de positions réalisent-elles cette distance ?
From Tatiana

$2^n - 1$; oui, par exemple, quand tous les disques doivent être posés à C

6. Donner un algorithme qui résoud les tours de Hanoï. Quel est sa complexité ?
From Tatiana

Quelques algos

Graphe non orienté

Définition 1. Connexité

Soit $G = (V, E)$ un graphe non orienté. G est *connexe*, si pour toute paire de sommets distincts $u \in V$ et $v \in V$, il existe une chaîne de u à v dans G.

Définition 2. Connexité au sens des arêtes

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

Définition 3. k-connexe

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

Exercice 5

Calculer $\lambda(G)$ (cf. definition 2) pour les graphes non orientés suivants :

— Un arbre ;
From Tatiana

1

— Chaîne de longueur $n \geq 1$;
From Tatiana

C'est aussi un arbre

— Cycle de longueur $n \geq 3$;
From Tatiana

2

— Graphe complet à $n \geq 2$ sommets ;
From Tatiana

$n - 1$

— Graphe complet biparti $K_{\ell,m}$ avec $\ell \geq 1$ et $m \geq 1$.
From Tatiana

$\min(l, m)$

Exercice 6

Exhiber un graphe non orienté 1-connexe sans sommet de degré 1. Sans sommet de degré k ?
From Tatiana

1) triangle-pont-triangle

2) $2K_n$ et un pont

Exercice 7

Soit G un graphe non orienté connexe et a une arête de G. Montrer que $\lambda(G) - 1 \leq \lambda(G - a) \leq \lambda(G)$. Ceci est-il vrai dans le cas orienté ?
From Tatiana

Trop facile : par example, par contradiction (si c'est plus petite, on a peut supprimer moins des arêtes que $\lambda(G)$ depuis le debut). Et cela ne peux pas être plus grand comme on peut supprimer les mêmes arêtes que de G et obtenir le graphe non-connexe.