Algorithmique des graphes

Cours 2 – Encore des définitions

František Kardoš

frantisek.kardos@u-bordeaux.fr

Plan

- Complexité des algorithmes
- Encore des définitions

Complexité algorithmique

Étant donné deux algorithmes différents pour résoudre le même problème, pour en choisir le meilleur on cherche des moyens de comparer leur performance.

Une des moyens de mesure l'efficacité d'un algorithme (un programme) est la *complexité*.

La complexité est définie comme la quantité de ressources (temps et/ou espace de mémoire) nécessaire pour résoudre un problème algorithmique.

Complexité algorithmique

Lorsque la complexité d'un algorithme est étudié, pour déterminer le temps de calcul nécessaire, on considère

- le nombre d'exécutions des opérations élémentaires (telles que affectations de valeur à une variable, opérations arithmétiques, tests de comparaison, appels à des fonctions, etc.)
- au pire cas
- en fonction de la taille de l'entrée
- de point de vue asymptotique

Quelques exemples

- recherche d'un sous-liste d'une somme donnée
- calcul du degré d'un sommet d'un graphe

Complexité – éléments de notation

La notation grand O de Landau (O comme "Ordnung") signifie intuitivement que une fonction ne croît pas plus vite qu'une autre.

Généralement, on exprime la complexité d'un algorithme $T(n)$ (le nombre précis d'opérations élémentaires) par une fonction générique $f(n)$ portant l'information sur l'ordre de grandeur de la vitesse de croissance de $T(n)$.

On note $T(n) = O(f(n))$ s'il existe des constantes N et c telles que

$$\forall n > N \quad T(n) \leq c \cdot f(n).$$

Exemples

$$3 + 5n + 4m = O(n + m)$$

$$2n^2 + 3n + 5 = O(n^2)$$

$$2n^2 + 1000000 = O(n^2)$$

$$2n^2 + 1000000n = O(n^2)$$

$$n^2 = O(n^3)$$

$$n^3 \neq O(n^2)$$

$$\sqrt{n} = O(n)$$

$$n \neq O(\sqrt{n})$$

$$\sqrt{3n + 1} = O(\sqrt{n})$$

D'autres notations

Une fonction $T(n)$ est en $O(f(n))$, si l'ordre de croissance de $f(n)$ est une borne supérieur sur l'ordre de croissance de $T(n)$. Plus précisément, la limite

$$\lim_{n \to \infty} \frac{T(n)}{f(n)}$$

existe et elle est finie.

D'autres notations

Une fonction $T(n)$ est en $O(f(n))$, si l'ordre de croissance de $f(n)$ est une borne supérieur sur l'ordre de croissance de $T(n)$. Plus précisément, la limite

$$\lim_{n\to\infty} \frac{T(n)}{f(n)}$$

existe et elle est finie.

Une fonction $T(n)$ est en $o(f(n))$, si l'ordre de croissance de $T(n)$ est strictement plus petit que celui de $f(n)$. Plus précisément,

$$\lim_{n\to\infty} \frac{T(n)}{f(n)} = 0.$$

D'autres notations

Une fonction $T(n)$ est en $\Omega(f(n))$, si l'ordre de croissance de $f(n)$ est une borne inférieur sur l'ordre de croissance de $T(n)$. Plus précisément,

$$\lim_{n \to \infty} \frac{T(n)}{f(n)} > 0.$$

D'autres notations

Une fonction $T(n)$ est en $\Omega(f(n))$, si l'ordre de croissance de $f(n)$ est une borne inférieur sur l'ordre de croissance de $T(n)$. Plus précisément,

$$\lim_{n\to\infty} \frac{T(n)}{f(n)} > 0.$$

Finalement, une fonction $T(n)$ est en $\Theta(f(n))$, si l'ordre de croissance de $f(n)$ est asymptotiquement le même que celui de $T(n)$. Plus précisément,

$$0 < \lim_{n\to\infty} \frac{T(n)}{f(n)} < \infty.$$

Encore des définitions

- Isomorphisme, sous-graphe (induit)
- Propriétés combinatoires des arbres
- Vocabulaire sur les graphes orientés

Isomorphismes des graphes

Quand deux représentations / matrices / dessins représentent en fait le même graphe ?

Isomorphismes des graphes

Quand deux représentations / matrices / dessins représentent en fait le même graphe ?

Soit G et H deux graphes simples. On dit que G et H sont *isomorphes* s'il existe une bijection $\varphi : V(G) \to V(H)$ telle que

$$uv \in E(G) \iff \varphi(u)\varphi(v) \in E(H).$$

Autrement dit, la seule différence entre G et H est (éventuellement) une permutation des sommets ; les deux représentent les mêmes relations parmi des objets.

Isomorphismes des graphes

Quand deux représentations / matrices / dessins représentent en fait le même graphe ?

Soit G et H deux graphes simples. On dit que G et H sont *isomorphes* s'il existe une bijection $\varphi : V(G) \to V(H)$ telle que

$$uv \in E(G) \iff \varphi(u)\varphi(v) \in E(H).$$

Autrement dit, la seule différence entre G et H est (éventuellement) une permutation des sommets ; les deux représentent les mêmes relations parmi des objets.

Exercice du cours : Comment définir isomorphisme pour des graphes pas forcément simples ?

Sous-graphe, sous-graphe induit

Soit $G = (V, E)$ un graphe. Un graphe $G' = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$ est un *sous-graphe* de G.

Sous-graphe, sous-graphe induit

Soit $G = (V, E)$ un graphe. Un graphe $G' = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$ est un *sous-graphe* de G.

Si $V(G') = V(G)$, on dit que G' est un sous-graphe *couvrant* de G.

Sous-graphe, sous-graphe induit

Soit $G = (V, E)$ un graphe. Un graphe $G' = (V', E')$ avec $V' \subseteq V$ et $E' \subseteq E$ est un *sous-graphe* de G.

Si $V(G') = V(G)$, on dit que G' est un sous-graphe *couvrant* de G.

Soit $G = (V, E)$ un graphe, soit $X \subseteq V$ un ensemble de sommets de G. Le sous-graphe de G induit par X est défini comme

$$G[X] = (X, \{uv \in E(G) : u, v \in X\}).$$

Chaînes (rappel)

Un *arbre* est un graphe connexe sans cycle.

Un graphe est *connexe* si tout sommet est accessible depuis n'importe quel autre sommet.

Deux sommets sont *accessibles* entre eux s'il existe une chaîne les reliant.

Chaînes (rappel)

Un *arbre* est un graphe connexe sans cycle.

Un graphe est *connexe* si tout sommet est accessible depuis n'importe quel autre sommet.

Deux sommets sont *accessibles* entre eux s'il existe une chaîne les reliant.

Une chaîne est *simple* si elle passe par toute arête au maximum une fois.

Une chaîne est *élémentaire* si elle passe par toute arête et par tout sommet au maximum une fois.

Chaînes

Une chaîne est *simple* si elle passe par toute arête au maximum une fois.

Une chaîne est *élémentaire* si elle passe par toute arête et par tout sommet au maximum une fois.

Lemme

S'il existe une chaîne reliant u et v dans un graphe G, alors il existe aussi une chaîne élémentaire les reliant.

Chaînes

Une chaîne est *simple* si elle passe par toute arête au maximum une fois.

Une chaîne est *élémentaire* si elle passe par toute arête et par tout sommet au maximum une fois.

Lemme

S'il existe une chaîne reliant u et v dans un graphe G, alors il existe aussi une chaîne élémentaire les reliant.

Preuve. Il suffit d'enlever des sous-chaînes entre des répétitions du même sommet.

Chaînes

Une chaîne est *simple* si elle passe par toute arête au maximum une fois.

Une chaîne est *élémentaire* si elle passe par toute arête et par tout sommet au maximum une fois.

Lemme

S'il existe une chaîne reliant u et v dans un graphe G, alors il existe aussi une chaîne élémentaire les reliant.

Preuve. Il suffit d'enlever des sous-chaînes entre des répétitions du même sommet.
Autre preuve. Considérons la chaîne la plus courte entre u et v. Elle est forcément élémentaire.

Chaînes dans des arbres

Un *arbre* est un graphe connexe sans cycle.

Lemme

Soit G un arbre. Pour toute paire de sommets u, v il existe une et une seule chaîne élémentaire les reliant.

Chaînes dans des arbres

Un *arbre* est un graphe connexe sans cycle.

Lemme

Soit G un arbre. Pour toute paire de sommets u, v il existe une et une seule chaîne élémentaire les reliant.

Preuve.

Chaînes dans des arbres

Un *arbre* est un graphe connexe sans cycle.

Lemme

Soit G un arbre. Pour toute paire de sommets u, v il existe une et une seule chaîne élémentaire les reliant.

Preuve.
L'existence d'une chaîne reliant u et v est garantie par la connexité de G.

Chaînes dans des arbres

Un *arbre* est un graphe connexe sans cycle.

Lemme

Soit G un arbre. Pour toute paire de sommets u, v il existe une et une seule chaîne élémentaire les reliant.

Preuve.
L'existence d'une chaîne reliant u et v est garantie par la connexité de G.
S'il existait deux chaînes élémentaires différentes reliant u et v, on trouverait dans le graphe G un cycle, une contradiction avec la supposition G étant un arbre.

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

Soit G un graphe avec n sommets et m arêtes. Reconstruisons G à partir d'un graphe G_0 sans arête en ajoutant les arêtes de G une par une.

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

Soit G un graphe avec n sommets et m arêtes. Reconstruisons G à partir d'un graphe G_0 sans arête en ajoutant les arêtes de G une par une.
G_0 se décompose en n composantes connexes – tout sommet est une composante connexe tout seul.

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

Soit G un graphe avec n sommets et m arêtes. Reconstruisons G à partir d'un graphe G_0 sans arête en ajoutant les arêtes de G une par une.
G_0 se décompose en n composantes connexes – tout sommet est une composante connexe tout seul.
En ajoutant une arête, soit le nombre de composantes connexes diminue de 1, soit un cycle est créé.

Lemme

Un graphe sans cycle a au maximum $n-1$ arêtes. S'il a exactement $n-1$ arêtes, il est forcément connexe.

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

Soit G un graphe avec n sommets et m arêtes. Reconstruisons G à partir d'un graphe G_0 sans arête en ajoutant les arêtes de G une par une.
G_0 se décompose en n composantes connexes – tout sommet est une composante connexe tout seul.
En ajoutant une arête, soit le nombre de composantes connexes diminue de 1, soit un cycle est créé.

Lemme

Un graphe connexe a au minimum $n-1$ arêtes. S'il a exactement $n-1$ arêtes, il est forcément sans cycle.

La taille d'un arbre

Un arbre à n sommets, combien peut-il avoir d'arêtes ?

Théorème

Soit G un graphe. Les conditions suivantes sont équivalentes :

1. *G est un arbre (c-à-d connexe et sans cycle) ;*
2. *G est connexe et $m = n - 1$;*
3. *G est sans cycle et $m = n - 1$;*
4. *G est connexe, en plus, la suppression de n'importe quelle arête le déconnecte ;*
5. *G est sans cycle, en plus, l'ajout de n'importe quelle arête crée un cycle.*

Graphes orientés

Un graphe orienté $G = (V, E)$ est un couple d'ensembles finis, dont

- V est l'ensemble de *sommets* de G, et
- E est l'ensemble d'*arcs* de G, où

tout arc relie un sommet à un (autre) sommet.

Si l'arc e relie u à v, on écrit $e = uv$, on dit que

- uv est un arc sortant de u,
- v est un voisin sortant / successeur de u,
- uv est un arc entrant à/de v,
- u est un voisin entrant / prédécesseur de v.

Graphes orientés – représentations

- Les listes de successeurs : pour chaque sommet du graphe la liste de ses successeurs ;
- La matrice d'adjacence : pour chaque paire de sommets v_i et v_j il est indiqué s'il existe un arc de v_i à v_j ;
- La matrice d'incidence : pour chaque sommet v_i et pour chaque arc e_j il est indiqué si l'arc e_j est un arc sortant (-1) ou entrant $(+1)$ du sommet v_i.

Graphes orientés : Degrés

Le *degré sortant* $deg^+(v)$ d'un sommet v est la longueur de la liste de successeurs de v.

Le degré sortant d'un sommet v est égal au nombre d'arcs sortants de v.

Graphes orientés : Degrés

Le *degré sortant* $deg^+(v)$ d'un sommet v est la longueur de la liste de successeurs de v.

Le degré sortant d'un sommet v est égal au nombre d'arcs sortants de v.

Le *degré entrant* $deg^-(v)$ d'un sommet v est la longueur de la liste de prédécesseur de v.

Le degré entrant d'un sommet v est égal au nombre d'arcs entrants de v.

Cheminement

Dans un graphe orienté G, un *chemin* de u à v est une séquence d'arcs consécutifs, par exemple $(u_0u_1, u_1u_2, \ldots, u_{k-1}u_k)$, telle que $u = u_0$ et $v = u_k$.

La *longueur* d'un chemin est le nombre d'arcs qui le composent (ici k).

Cheminement

Dans un graphe orienté G, un *chemin* de u à v est une séquence d'arcs consécutifs, par exemple $(u_0u_1, u_1u_2, \ldots, u_{k-1}u_k)$, telle que $u = u_0$ et $v = u_k$.

La *longueur* d'un chemin est le nombre d'arcs qui le composent (ici k).

On dit qu'un sommet v est *accessible* à partir du sommet u s'il existe un chemin de u à v.

Un graphe orienté est dit *fortement connexe* si ses sommets sont tous accessibles entre eux, i.e., pour toute paire de sommets u et v il existe un chemin de u à v et aussi un chemin de v à u.

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

À partir de la matrice d'adjacence :

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

À partir de la matrice d'adjacence :
Il suffit de calculer la somme de la colonne du sommet v.

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

À partir de la matrice d'adjacence :
Il suffit de calculer la somme de la colonne du sommet v.

À partir de la matrice d'incidence :

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

À partir de la matrice d'adjacence :
Il suffit de calculer la somme de la colonne du sommet v.

À partir de la matrice d'incidence :
Il suffit de calculer le nombre de $+1$ dans la ligne de v.

Exercice

Pour chacune des trois représentations, concevoir un algorithme qui, étant donné un graphe orienté G et un sommet v de G, calcule le degré entrant de v.

À partir des listes de successeurs :
Il faut parcourir toutes les listes et compter le nombre total d'occurrences de v dans l'ensemble des listes.

À partir de la matrice d'adjacence :
Il suffit de calculer la somme de la colonne du sommet v.

À partir de la matrice d'incidence :
Il suffit de calculer le nombre de $+1$ dans la ligne de v.

Quelle est la complexité ?