Calcul de triangles

Samy Jaziri

Sujet de khôlle inspiré du sujet d'oral d'informatique de l'ENS ULM 2018

Préambule

*Les candidats sont libres d'utiliser le **C** ou le **OCaml** pour coder les algorithmes de ce sujet.*

Pour vous mettre dans les conditions du concours, vous n'avez pas le droit d'accéder aux ressources en ligne.

En ce qui concerne les questions orales de la khôlle, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de fa'c'on schématique, courte et précise. Vous ne devez pas expliquer votre code ligne par ligne! Quand la complexité d'un algorithme est demandée en temps ou en mémoire en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, donné en notation de Landau ($\mathcal{O}(n), \mathcal{O}(\log(n)), \ldots$). Prenez des notes lorsque vous préparez une question orale pour retrouver plus rapidement les grandes lignes de votre explication lorsque l'examinateur passe vous voir.
Il vous est **demandé** de **tester vos programmes sur des petits exemples** que vous aurez résolus préalablement à la main.
Vous devrez tester votre programme sur des exemples bien choisis permettant de vérifier rapidement que votre code est fonctionnel.
Il vous est demandé d'aborder les questions dans l'ordre et de noter vos difficultés à répondre à une question avant de passer à la suivante. Vous pourrez alors les aborder avec l'examinateur.

1 Mise en jambe

On considère un graphe non-orienté $G = (V, E)$ où $V = \{1, \ldots, n\}$ est l'ensemble des sommets et E est un ensemble d'arêtes qui sont des paires de sommets de V. Un *triangle* dans G est un sous-ensemble $\{x, y, z\} \in V$ tel que $\{x, y\}, \{y, z\}$, et $\{x, z\}$ appartiennent à E. Dans ce problème, on veut concevoir des algorithmes qui prennent en entrée un graphe et calculent (sans doublons) l'ensemble des triangles du graphe.

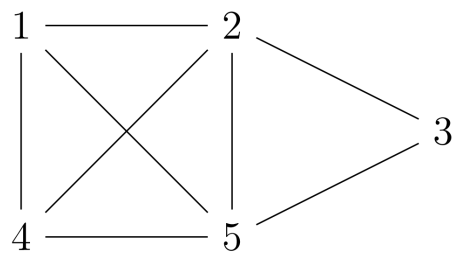

Figure 1: Graphe Γ

Préparer une réponse à donner à l'oral

Déterminer les triangles du graphe Γ donné en illustration.

Question 1

Étant donné un graphe G *représenté comme une matrice d'adjacence*, implémenter un algorithme naïf en $\mathcal{O}(|V|^3)$ pour déterminer l'ensemble des triangles de G.

2 Algorithmes alternatifs

Dans la suite du sujet, on supposera toujours que le graphe d'entrée est fourni sous forme de listes d'adjacence, et on supposera toujours que chacune de ces listes est triée.

Préparer une réponse à donner à l'oral

Donner un algorithme (en pseudo-code) calculant l'intersection de deux listes triées en temps linéaire.

Question 2

Étant donné un graphe G, implémenter un algorithme en $\mathcal{O}(|E| \times |V|)$ pour déterminer l'ensemble des triangles de G. Tester votre programme.

Préparer une réponse à donner à l'oral

Si l'on compare l'algorithme de la question 1 et celui de la question 2, lequel parmi ces algorithmes a la meilleure complexité ?
Le choix de la représentation du graphe d'entrée était-il important ?

Question 3

Étant donné un graphe G, si l'on note Δ le degré maximal d'un sommet de G, implémenter un algorithme en $\mathcal{O}(|E| \times \Delta)$ pour déterminer l'ensemble des triangles de G. Tester votre programme.

Préparer une réponse à donner à l'oral

Commenter la performance de cet algorithme.

3 Algorithme Heavy-Light

On dit qu'un sommet de G est lourd si son degré est supérieur ou égal à $\sqrt{|E|}$.

Préparer une réponse à donner à l'oral

Montrer que si $H \subseteq V$ est l'ensemble des sommets lourd de G, $|H| \leq 2\sqrt{|E|}$.

Préparer une réponse à donner à l'oral

Expliquer comment calculer en temps linéaire en $|E|$, pour chaque sommet de G la liste de ses voisins lourds et de ses voisins léger. Les listes produites seront toujours triées.

On propose de traiter différemment les sommets lourd et les sommets légers dans l'algorithme. On calcule d'abord tous les triangles contenant au moins un sommet lourd, en utilisant un algorithme similaire à celui de la question 2. Puis on calcule les triangles ne contenant que des sommets légers avec un algorithme similaire à celui de la question 3.

Préparer une réponse à donner à l'oral

Donner en pseudo code l'algorithme final et montrer que sa complexité est en $\mathcal{O}(|E|^{\frac{3}{2}})$

Question 4

Implémenter l'algorithme. Tester votre programme.