Cycles

Samy Jaziri

Sujet de khôlle inspiré par les « Graph Problems » du livre The Algorithme Design Manual *de Steven S. Skiena*

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 $(\mathcal{O}(n), \mathcal{O}(\log(n)), \dots)$. 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 *cycle* ou *cycle simple* dans G est un chemin dans G dont le départ et l'arrivée sont identiques et dont toutes les arêtes sont différentes.
Dans la suite du sujet, on supposera toujours que le graphe d'entrée est fourni sous forme de listes d'adjacence.

Question 1

Étant donné un graphe G, implémenter un algorithme qui renvoie un cycle dans G s'il existe.

2 Cycles Eulériens

Un cycle est dit *Eulérien* s'il passe exactement une fois par chaque arête de G. On suppose dans la suite que G est un graphe **connexe**.

> *Préparer une réponse à donner à l'oral*
>
> Montrer que G possède un cycle Eulérien implique que chaque sommet de G est de degré pair.

Question 2

Implémenter une fonction qui, étant donné un graphe G et un cycle C de G, vérifie si C est un cycle Eulérien de G.

> *Préparer une réponse à donner à l'oral*
>
> Deux cycles de G sont dit *disjoints* s'ils ne partagent aucune arête. Expliquer, étant donnés deux cycles disjoints de G qui possèdent un sommet commun, comment construire un cycle de G regroupant les arêtes de ces deux cycles.

Soit C un cycle de G. Supprimer C de G consiste à éliminer toutes les arêtes de C dans G.

> *Préparer une réponse à donner à l'oral*
>
> On suppose que chaque sommet de G est de degré pair. Montrer qu'après la suppression d'un nombre arbitraire de cycles dans G, G ne possède plus d'arête ou G possède un nouveau cycle.

Question 3

Implémenter une fonction qui supprime un cycle de G et le renvoie.

Préparer une réponse à donner à l'oral

On suppose que G est connexe et chaque sommet de G est de degré pair. Donner un algorithme en pseudo-code qui construit une liste de cycles disjoints de G contenant toutes les arêtes de G. Donner en quelque ligne, la preuve de sa correction, de sa terminaison et l'analyse de sa complexité.

Préparer une réponse à donner à l'oral

En déduire une caractérisation des graphes connexies non-orientés contenant un cycle Eulérien.

Question 4

Implémenter un algorithme linéaire renvoyant un cycle Eulérien de G s'il existe. Tester votre programme.

3 Cycles Hamiltoniens

Un cycle est dit *Hamiltonien* si il passe exactement une fois par chaque sommet de G.

Question 5

Implémenter une fonction qui renvoie un cycle Hamiltonien de G s'il en existe un. Tester votre algorithme.
Savoir, étant donné un graphe G, s'il possède un cycle Hamiltonien est un problème NP-complet. Nous nous satisferons donc d'un algorithme en $\mathcal{O}(2^{|E|})$.