Sélection internationale Session 2014
École Normale Supérieure Paris
Épreuve de culture scientifique - **Informatique** Durée : 3 heures

Pour les candidats ayant choisi **l'informatique** comme **spécialité principale**

Si vous ne parvenez pas à répondre à une question, vous pouvez cependant l'utiliser comme hypothèse pour les questions suivantes.

Calculatrices interdites.

Exercice 1.

1. Décrire une structure de données S pour stocker des éléments pondérés qui permet les opérations suivantes.

   - Insérer(S, elem, w) ajoute un élément de poids w à la structure S,
   - Recherche-min(S) retourne un élément de poids minimum dans S ainsi que son poids,
   - Efface-min(S) enlève l'élément de poids minimum de S.

   Le temps de calcul de Insérer et Efface-min doivent être $O(\log n)$ où n est le nombre d'éléments dans S au moment de l'appel. Recherche-min doit fonctionner en $O(1)$.

   Vous pouvez supposer que les poids sont entiers.

   Donner un argument bref expliquant que vos fonctions se déroulent dans les temps souhaités et produisent des résultats corrects.

2. Combien de temps faut-il pour aller de la ville a à la ville z si le temps de voyage entre les villes sont les suivantes?

   - 1 minutes entre les villes a et b
   - 9 minutes entre les villes a et c
   - 4 minutes entre les villes a et d
   - 5 minutes entre les villes b et e
   - 12 minutes entre les villes b et f
   - 12 minutes entre les villes b et f
   - 4 minutes entre les villes c et f
   - 4 minutes entre les villes c et g
   - 2 minutes entre les villes d et g
   - 8 minutes entre les villes e et h
   - 3 minutes entre les villes f et h
   - 7 minutes entre les villes f et z
   - 2 minutes entre les villes f et i
   - 3 minutes entre les villes h et z

- 6 minutes entre les villes i et z

3. Utilisant le langage de programmation de votre choix, écrire un programme qui prends en entrée des entiers décrivant le temps de voyage entre deux villes et calcule le temps minimum qu'il faut pour aller de la ville 1 à la ville 2. Chaque ville est représenté par un numéro entre 1 et 1000000.

   Le temps de calcul de votre programme devrait être $O((m+n)\log m)$ où m est le nombre de paires de villes donnés à l'entrée et n le nombre de villes.

   Supposer que vous avez accès à la structure décrit dans la Question 1.

4. Maintenant, supposons que traverser une ville (c'est à dire arriver et repartir) prends aussi du temps. Le temps requis est la moitié du nombre de minutes restant jusqu'à la prochaine heure. Donc, si vous arrivez à 2:16 (soit 136 minutes après le départ), vous sortez de la ville dans $44/2 = 22$ minutes. Arriver pile à l'heure vous laisser sortir immédiatement.

   Décrire un algorithme pour trouver le temps minimum requis pour arriver à la ville z de la ville a si vous commencez à a à minuit (i.e., 00:00).

5. Décrire comment le temps de calcul d'une des opérations Insérer or Efface-min de la Question 1 peut être améliorée en gardant le temps de calcul des deux autres opérations.

Exercice 2.

Un *graphe* G est un paire ordonnée (V, E) où V est un ensemble appelé sommets et E est un sous ensemble des paires (non-ordonnées) de V appelé arêtes. Deux sommets $u, v \in V$ sont *adjacent* si (u, v) est un élément de E. Le *degré d'un sommet* u est le nombre d'arêtes contenant u.

Un *chemin* dans un graphe G est une suite de sommets $v_1, v_2, \ldots, v_k$ où v_i et v_{i+1} sont adjacents pour tout i and aucun sommet n'apparaît deux fois dans cette suite. Un *cycle* dans un graphe G est un chemin où v_1 et v_k sont adjacent. Un *cycle Hamiltonian* dans un graphe G est un cycle qui contient tout sommet de G.

Le but de cet exercice est de donner un algorithme polynomial pour trouver un cycle Hamiltonian dans un graphe $G = (V, E)$ avec au moins trois sommets et où tous les sommets ont degré au moins $|V|/2$. Un graphe avec ces propriétés est un *graphe de degré élevé.*

Vous pouvez directement donner un tel algorithme et démontrer qu'il est correcte et a un temps de calcul polynomial à la place de répondre aux Questions 1-5. Dans ce cas, vous devez toujours répondre à la Question 6.

1. Donner un algorithme polynomial qui prends en entrée un graphe de degré élevé G et chemin $P = v_1, \ldots, v_k$ dans G où $k < n$ et retourne soit un chemin avec $k + 1$ sommets ou un cycle contenant les sommets de P (possiblement dans un ordre différent).

2. Donner un algorithme polynomial qui prends en entrée un graphe de degré élevé G et cycle C dans G où $k < n$ et retourne un chemin avec $k + 1$ sommets contenant tous les sommets de C (possiblement dans un ordre différent).

3. Donner un algorithme polynomial qui prends en entrée un graphe de degré élevé G et retourne un cycle Hamiltonian dans G.

4. Montrer que le temps de calcul de votre algorithme de la Question 3 est bien polynomial. Écrire explicitement la valeur de k pour lequel votre algorithme a un temps de calcul $O(n^k)$.

5. Montrer que votre algorithme de la Question 3 est correcte. C'est à dire, montrer qu'il retourne bien un cycle et que celui-ci est Hamiltonian. (Selon votre algorithme, il est peut être nécessaire de montrer qu'il ne bloque pas et termine.)

6. Expliquer comment changer l'algorithme de la Question 3 en un algorithme qui prends en entrée un graphe $G = (V, E)$ avec au moins 3 sommets et où la somme des degrés de n'importe quel paire de sommets non-adjacents u, v est au moins $|V|$ et qui retourne un cycle Hamiltonian dans G.