IFT436 – Algorithmes et structures de données

Université de Sherbrooke

Devoir 3

Enseignant:	Michael Blondin
Date de remise:	jeudi 10 octobre 2019 à 10h30
À réaliser:	en équipe de deux ou individuellement
Modalités:	à remettre en classe, en copie imprimée ou copie manuscrite lisible
Bonus:	les questions bonus sont indiquées par ★
Pointage:	max. 50 points + 3 points bonus

Question 1.

Vous passez la journée dans un parc aquatique des Cantons-de-l'Est. L'une des attractions comporte n bassins d'eau qu'on descend en tube via des corridors aquatiques qui respectent les propriétés suivantes:

— Il existe *au plus un* corridor aquatique passant d'un bassin i vers un autre bassin j;

— Il n'y a *pas* de corridor passant d'un bassin vers lui-même;

— Puisque les bassins sont situés de plus en plus bas le long d'une colline, si on peut atteindre un bassin j à partir d'un bassin i, alors on ne peut *pas* atteindre le bassin i à partir du bassin j;

— On peut atteindre tous les bassins à partir du bassin de départ, et on peut atteindre le bassin d'arrivée à partir de tous les bassins;

— S'il y a un corridor du bassin i vers le bassin j, alors le temps pour le traverser en tube est d'exactement $c[i, j] > 0$ minutes.

Comme la file d'attente est incroyablement longue, vous planifiez maximiser votre temps dans l'attraction. Vous remarquez que, de façon générale, une telle attraction peut être modélisée comme un graphe $\mathcal{G} = (V, E)$.

(a) Dites à quoi correspondent V et E, et si $\mathcal{G}$ est *dirigé* ou non. 2 pts

(b) Dites si $\mathcal{G}$ peut contenir un *cycle* (non vide). Justifiez brièvement. 2 pts

(c) Dites si $\mathcal{G}$ peut être un *arbre*. Si $\mathcal{G}$ est dirigé, considérez-le sans ses directions. Justifiez brièvement. 2 pts

(d) Dites combien d'arêtes $\mathcal{G}$ peut avoir au *minimum* et au *maximum* par rapport à n. Justifiez brièvement. 2 pts

(e) Dites comment déterminer le bassin de *départ* et le bassin d'*arrivée* de $\mathcal{G}$. Justifiez brièvement. 3 pts

(f) **Donnez un algorithme qui détermine le *temps maximal* que vous pouvez passer dans une attraction** $\mathcal{G}$. Analysez le temps d'exécution de votre algorithme. Afin d'obtenir tous les points, il doit fonctionner en temps $\mathcal{O}(n + m)$, où m est le nombre de corridors aquatiques. 8 pts

Considérez l'entrée du problème comme un graphe $\mathcal{G}$ représenté par une liste d'adjacence, ainsi qu'un tableau associatif $c\colon [n] \times [n] \to \mathbb{N}_{>0}$ tel que l'évaluation de $c[i, j]$ se fait en temps constant.

Indice: considérez les propriétés de $\mathcal{G}$, et pensez au temps maximal en partant de bassins intermédiaires.

(g) Supposez maintenant qu'il y ait un sentier pédestre de chaque bassin vers le bassin de départ. Le temps afin de remonter du bassin i vers le bassin de départ est d'exactement $r[i] > 0$ minutes. Vous pouvez remonter au plus 5 fois par l'un des sentiers avant qu'un·e employé·e le remarque. **Adaptez votre algorithme afin qu'il calcule le *temps maximal sous ces contraintes*.** Afin d'obtenir tous les points, il doit fonctionner en temps $\mathcal{O}(n + m)$. Considérez r comme un tableau associatif $r\colon [n] \to \mathbb{N}_{>0}$ tel que l'évaluation de $r[i]$ se fait en temps constant. 4 pts

Question 2.
Soit s une séquence d'éléments comparables. Nous écrivons $|s|_x$ afin de dénoter le nombre d'occurrences de x dans s. Un *mode* m de s est une valeur qui apparaît un nombre maximal de fois dans s; autrement dit, tel que $|s|_m = \max\{|s|_x : x \in s\}$. Par exemple, $s = [42, 42, 0, 1, 0, 42, 0]$ possède deux modes: 0 et 42.

(a) Adaptez l'algorithme de tri rapide afin de résoudre ce problème en temps $\mathcal{O}(n \log n)$: 8 pts

ENTRÉE: une séquence s de $n \in \mathbb{N}_{>0}$ éléments comparables
SORTIE: un mode m de s

Vous n'avez *pas* à analyser le temps d'exécution de votre algorithme.

Vous avez accès à une instruction `médiane`(s) qui retourne la médiane d'une séquence s de $n > 0$ éléments en temps $\mathcal{O}(n)$. Dans ce contexte, la médiane réfère à la valeur qui se retrouve à la position $\lceil n/2 \rceil$ lorsque s est triée; par ex. la médiane de $[20, 40, 10, 50, 60, 30]$ est 30 et la médiane de $[70, 20, 40, 10, 50, 60, 30]$ est 40. Ainsi, pour les séquences de taille paire, on ne prend *pas* la moyenne des deux valeurs milieux comme en statistiques.

Remarque: on pourrait aussi trouver un mode en triant d'abord, puis en effectuant un parcours linéaire; mais ce n'est pas ce qui est demandé, nous cherchons une approche récursive qui résout directement le problème sans prétraitement.

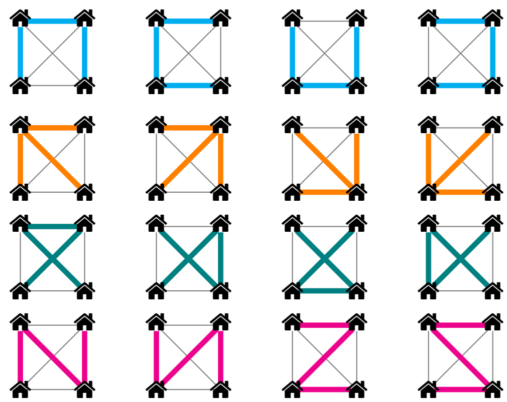
(b) Donnez une trace de l'arbre de récursion de votre algorithme sur entrée: 4 pts

$$s = [10, 20, 20, 20, 10, 30, 70, 20, 80, 30, 60, 40, 20, 10, 50, 10, 70, 80].$$

Question 3.
Une entrepreneure compte aménager un site de glamping écoresponsable de n gîtes dans les Cantons-de-l'Est, où n sera déterminé selon la subvention accordée par le Ministère de l'Environnement et de la Lutte contre les changements climatiques. L'entrepreneure désire qu'on puisse marcher de chaque gîte vers chaque autre gîte à partir d'un réseau de sentiers. Elle vous embauche donc afin de l'épauler dans la conception du site.

(a) Étant donné un réseau de sentiers, on vous demande de vérifier rapidement s'il y a un chemin entre chaque paire de gîtes, mais pas de chemin superflu. Autrement dit, **donnez un algorithme qui détermine si un graphe non dirigé est un *arbre***. Expliquez brièvement pourquoi l'algorithme fonctionne, et analysez son temps d'exécution. Votre algorithme doit fonctionner en temps $\mathcal{O}(n)$ dans le pire cas, où n est le nombre de sommets. Considérez l'entrée comme un graphe représenté par une liste d'adjacence. 6 pts

(b) L'entrepreneure aimerait que vous produisiez les plans de *tous* les réseaux de sentiers qui connectent les gîtes sans chemin superflu. Vous considérez $n = 4$ et remarquez qu'il y a $2^4 = 16$ réseaux possibles: 6 pts



Le croisement au centre n'est pas physique, imaginez plutôt un pont au-dessus d'un sentier.

Comme n risque d'être grand, vous vous inquiétez quant à la charge de travail. Vous aimeriez donc convaincre l'entrepreneure qu'il y a beaucoup trop de plans possibles. Pour ce faire, **démontrez que le graphe complet $\mathcal{G}_n$ possède au moins 2^n arbres couvrants, pour tout $n \in \mathbb{N}_{\geq 4}$.**

Indice: pensez à une façon d'étendre un arbre couvrant de $\mathcal{G}_n$ à un arbre couvrant de $\mathcal{G}_{n+1}$.

Rappels:

— Le *graphe complet* $\mathcal{G}_n$ est le graphe non dirigé de n sommets qui possède une arête entre chaque paire de sommets. Autrement dit, $\mathcal{G}_n$ correspond au réseau où il y a un sentier entre chaque paire de gîtes.

— Un *arbre couvrant* d'un graphe $\mathcal{H}$ est un sous-graphe de $\mathcal{H}$ qui est un arbre et qui contient tous les sommets de $\mathcal{H}$.

(c) L'entrepreneure aimerait maintenant minimiser la distance du réseau. Vous désirez la convaincre qu'il peut être préférable d'ajouter une halte entre les gîtes. Plus formellement, considérez les gîtes comme n points dans le plan, et un chemin entre deux gîtes comme une droite dont la longueur correspond à la distance euclidienne entre les deux points. **Donnez un exemple de points où un arbre couvrant minimal entre ces points est *plus long* qu'un arbre couvrant minimal avec un nouveau point ajouté (une halte)**. Choisissez le nombre de gîtes, ainsi que leur position et celle de la halte. Justifiez. 3 pts

★ Donnez un algorithme qui résout le problème suivant en temps $\mathcal{O}(k)$: ★ 3 pts

ENTRÉE: une forêt $\mathcal{F} = (V, E)$ sous forme de liste d'adjacence, et un entier $1 \leq k < |V|$, tels que $\deg(v) > 0$ pour tout $v \in V$

SORTIE: $\mathcal{F}$ contient-elle au moins $|V| - k$ arbres?

Considérez l'arithmétique, et ces opérations sur les séquences, comme élémentaires: l'ajout, l'accès, et l'obtention de la taille. Analysez le temps d'exécution de l'algorithme et expliquez pourquoi il est correct.