IFT 2125 Automne 2017 **Pierre McKenzie**
Introduction à l'algorithmique

Devoir 2

Remise : le mercredi 11 octobre au *début* de la démo
Notez : à rendre sous forme *papier*

1. Le "lemme des puissances de b", page 15 du deuxième jeu de transparents, comprend trois cas. Nous avons démontré les cas 1 (en démo) et 3 (en cours). Démontrez le cas 2.

   The "powers of b lemma", on slide 15 of the second batch of slides, splits into three cases. We have proved cases 1 (Maëlle) and 3 (in class). You are asked to prove case 2.

2. Découvrez une fonction simple $f(n)$ telle que $t_n \in \Theta(f(n))$, en vous servant de la méthode de l'équation caractéristique et en décrivant vos étapes.

   Find a simple function $f(n)$ such that $t_n \in \Theta(f(n))$, using the characteristic method and showing your steps.

$$t_n = \begin{cases} 2 & \text{si } n = 0, \\ 29 & \text{si } n = 1, \\ 12t_{n-1} - 35t_{n-2} - 6 \cdot 5^{n-1} & \text{si } n > 1. \end{cases}$$

3. Votre voiture électrique a une autonomie de 300 kilomètres et vous planifiez un trajet de Montréal à Vancouver. Vous connaissez les positions $s_1, \ldots, s_n$ des postes de recharge le long du trajet, qui sont à une distance d'au plus 300 kilomètres de l'un au suivant. Décrivez un algorithme vorace qui minimisera le nombre d'arrêts le long de votre trajet, et démontrez ensuite (vraisemblablement par induction) que votre algorithme fonctionne.

   You have planned a route from Montreal to Vancouver. Your electric car can travel 300 km between recharges. Recharging stations along the route are at positions $s_1, \ldots, s_n$, separated from one to the next by at most 300 km. Describe an algorithm using a greedy strategy that will minimize the number of recharging stops, and prove formally (likely by induction) that your algorithm works.

4. Rappelons qu'un *stable* d'un graphe non dirigé est un ensemble de sommets dont aucune paire n'est reliée par une arête.

   Recall that a "stable" of a graph, or "independent set" in English, is a set of nodes that contains no adjacent pair of nodes.

   (a) Soit $G = (S, A)$ un graphe. Est-ce que $\{S, \{I \subseteq S : I \text{ est un stable }\}\}$ forme un matroïde ? (Justifiez.)

   Let a graph $G = (S, A)$ be given. Does $\{S, \{I \subseteq S : I \text{ is a "stable" }\}\}$ form a matroid? (Justify.)

   (b) Décrivez en mots une stratégie vorace visant à résoudre le problème *Stable* du transparent 14 de l'introduction au cours et implantez votre méthode en Python. Vous pouvez faire appel à une méthode pour faire le tri sans la programmer vous-même si vous le souhaitez. Déposez votre méthode sur Studium.

   (Supposez que le graphe en entrée vous est fourni sous forme de matrice d'adjacence, comme dans le fichier graphM.py accompagnant l'énoncé de ce devoir sur Studium.)

   Describe in words a greedy strategy attempting to solve the problem Stable from slide 14 in the introduction to the course, and implement your method in Python. You may appeal to a sorting method in Python without having to program it yourself if you wish. Upload your method on Studium. (The input graph is provided to your method by an adjacency matrix, such as the graphM.py file on Studium.)

(c) Si votre méthode résout *Stable*, justifiez. Sinon, exhibez sur papier, à la main, un petit exemplaire de *Stable* sur lequel votre méthode échoue.

If your method solves Stable, then justify. Otherwise give by hand, on paper, a small size instance of Stable on which your method fails.