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

Devoir 4 (partiel : la question 4 est à venir ; question 4 will come next week)
Remise : le mercredi 6 décembre (au *début* de la démo)

1. Faites le problème 8.32 de B&B (le canoteur économe).

   Do the problem 8.32 in the Brassard and Bratley book.

2. Une application $\rho : A \times A \to A$ est fixée une fois pour toutes, où A est l'ensemble $\{a, b, c, \ldots, y, z\}$ des 26 lettres de l'alphabet. Donnez un algorithme utilisant la technique de la programmation dynamique et résolvant le problème suivant :

   <u>Évaluation</u>
   DONNÉE: $\sigma_1, \ldots, \sigma_n \in A$
   DÉCIDER: s'il existe un parenthésage complet de $\sigma_1 * \sigma_2 * \cdots * \sigma_n$ qui permet, en remplaçant à répétition $(\alpha * \beta)$ où $\alpha \in A$ et $\beta \in A$ par $\rho(\alpha, \beta)$ dans un ordre prescrit par le parenthésage, d'obtenir tout simplement à la fin la lettre a.

   Indice. Pensez à un tableau dont chaque entrée est un ensemble de lettres de A.

   A total function $\rho : A \times A \to A$ is specified once and for all, where $A = \{a, b, c, \ldots, y, z\}$. Give a dynamic programming algorithm that solves the following problem :

   <u>Évaluation</u>
   DONNÉE: $\sigma_1, \ldots, \sigma_n \in A$
   DÉCIDER: if there exists a complete bracketing of $\sigma_1 * \sigma_2 * \cdots * \sigma_n$ that leaves a in the end when $(\alpha * \beta)$ for $\alpha, \beta \in A$ is systematically replaced with $\rho(\alpha, \beta)$ according to the bracketing.

   Hint. Think of a table with subsets of A as entries.

3. Donnez en Python (et imprimez papier et placez sur Studium) un algorithme de retour arrière qui résout le problème suivant :

   <u>MêmeGraphe</u>
   DONNÉE: matrices d'adjacence symétriques $A, B \in \{0, 1\}^{m \times m}$
   DÉCIDER: si A et B représentent le même graphe non orienté.

   Indice. A et B représentent le même graphe si et seulement si une permutation σ de l'ensemble $\{1, 2, \ldots, m\}$ existe telle que pour tout $i, j \in \{1, 2, \ldots, m\}$, $A(i, j) = B(\sigma(i), \sigma(j))$.

   Give (and print and upload on Studium) a Python backtracking algorithm to solve :

   <u>SameGraph</u>
   DONNÉE: symmetric adjacency matrices $A, B \in \{0, 1\}^{m \times m}$
   DÉCIDER: if A and B represent the same undirected graph.

   Hint. A and B represent the same graph iff a permutation σ of the set $\{1, 2, \ldots, m\}$ exists such that for all $i, j \in \{1, 2, \ldots, m\}$, $A(i, j) = B(\sigma(i), \sigma(j))$.