

# 8INF259: STRUCTURES DE DONNÉES

## TP2 : Arborescence génétique

### Travail seul ou en équipe de 2

À remettre au plus tard le **mardi 2 avril 2019 à 23 h 55**

#### 1. Objectif

Les objectifs de ce devoir sont:

- L'implémentation d'une structure d'arbre répondant à des besoins spécifiques;
- L'implémentation et l'utilisation de patrons de classe (template);
- La surcharge d'opérateurs (opérateurs "+", "<<" et ">>");
- L'utilisation des libraires standard de C++ lorsque nécessaire;

#### 2. Mise en contexte

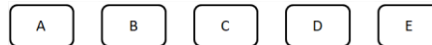
Arbotic inc., une nouvelle compagnie de logiciel, travaille à la mise en œuvre d'une application qui permet aux chercheurs généticiens de générer un individu issu de différentes cellules d'une même espèce.

Le programme, à terme, permettra de tester différents types de cellules, comportant chacune leurs particularités, tant pour ce qui est de leur mécanisme de reproduction que de la composition de leur bagage génétique. Quoi qu'il en soit, le fonctionnement pour la génération des individus reste toujours le même.

Au départ, l'algorithme demande une liste d'individus. Pour générer l'individu final, deux individus sélectionnés au hasard sont retirés de la liste et additionnés entre eux. Le résultat, également un individu, est ensuite ajouté à la liste d'individus. Cette opération se fait tant qu'il y a plus d'un élément dans la liste. Dans l'exemple ci-dessous, "I" est l'individu résultant.

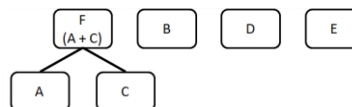
---

Population initiale :



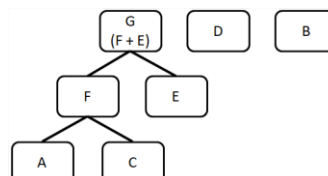
---

Après une itération :



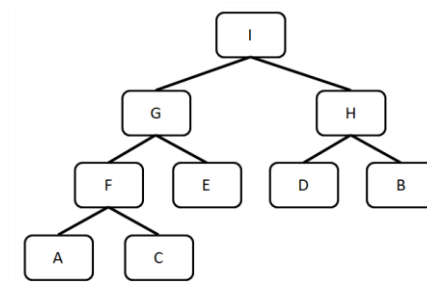
---

Après deux itérations :



---

Résultat final:



---

### 3. Travail à effectuer

On s'intéresse à différents aspects de la généalogie de l'individu:

- Les caractéristiques de l'individu résultant;
- Le nombre de générations qui le sépare de la population initiale;
- Les caractéristiques de tous ses ancêtres d'un niveau donné;
- Afficher l'arborescence génétique complète de l'individu à l'aide de l'opérateur "<<";

Vous devez donc implémenter un patron de classe "GenTree" permettant de créer, à partir d'une liste d'individus, un arbre génétique tel qu'expliquer précédemment. Chacun des individus doit pouvoir s'additionner (operator+). De plus, on doit pouvoir lire chaque individu avec l'opérateur ">>". On doit également pouvoir envoyer le GenTree dans un flux de sortie à l'aide de l'opérateur "<<". Le résultat doit montrer la généalogie de l'individu résultant sous forme d'arbre.

Vous devez également tester votre structure avec deux types de classe :

- Pour des cellules de type "ColorCell";
  - Possède trois entiers indiquant les composantes (RGB)
  - L'addition de deux cellules  $C_1$  et  $C_2$  crée une nouvelle cellule  $C_3$  ayant comme valeur

$$R_3 = \frac{R_1 + R_2}{2}, \quad G_3 = \frac{G_1 + G_2}{2}, \quad B_3 = \frac{B_1 + B_2}{2}$$

- Une cellule est représentée textuellement par une ligne composée de trois entiers entre 0 et 255.

Par exemple: 100 255 0

- Pour des cellules de type "ChromoCell"
  - Possède quatre paires de chromosomes (paire1, paire2, paire3, paire4), chacune identifiée par une lettre majuscule de A à L;
  - L'addition de deux cellules  $C_1$  et  $C_2$  crée une nouvelle cellule  $C_3$  où chaque paire de chromosomes est formée par la sélection aléatoire d'un chromosome de la paire correspondante de chacun de ses parents.

Par exemple, si  $C_1$  (AA BB CC DD) s'additionne à  $C_2$  (EE FF GG HH), le résultat pourrait être  $C_3$  (AE FB GC HD).

- Une cellule est représentée textuellement par une ligne composée de quatre couples de deux lettres, séparés par un espace. Par exemple: AB DL EA CD

#### 4. Fichiers de départ

Vous devez utiliser comme point de départ la solution présentée sur Moodle.

**À l'exception du fichier main.cpp qui doit rester inchangé, vous pouvez changer les fichiers comme vous le désirez.**

La première ligne des fichiers de population sert à identifier le type de cellule ("ChromoCell" ou "ColorCell"). Par la suite, chaque ligne du fichier représente une cellule.

**Des exemples de fichiers de population seront disponibles sur Moodle.**

#### 5. Exigences et livrables

Afin de simplifier la correction du devoir, vous devez respecter ces quelques consignes:

- Votre devoir doit obligatoirement fonctionner (compilation et exécution) avec l'environnement de programmation Visual Studio 2015 ou 2017.
- Vous devez remettre tous vos fichiers d'entêtes et vos fichiers sources (.h et .cpp);
- Vous pouvez remettre la structure de projet contenant les fichiers .sln et .vxproj. Cependant, vous devrez vous assurer d'avoir nettoyé votre solution avec la remise (seulement les fichiers .sln, .vxproj, .h et .cpp).
- Vous devez remettre un seul fichier .zip contenant les fichiers de votre devoir. Ce fichier doit indiquer le code permanent des membres de l'équipe (ex.: **ETUA01234567\_ETUB01234567.zip**)
- Vous devez déposer votre devoir via le site Moodle du cours dans la section prévue à cet effet.

## 6. Guide de correction

Présenté à titre indicatif seulement

<b>Guide de correction</b>	
<b>Patrons de classe</b>	
<b>Patron de classe GenTree</b>	/4
<b>Formation d'un GenTree à partir d'une population initiale</b>	
<b>Classes de Cellules</b>	
<b>ColorCell</b>	/1
<b>ChromoCell</b>	
<b>Surcharge des opérateurs</b>	
<b>Opérateur +</b>	/3
<b>Opérateur &lt;&lt;</b>	
<b>Opérateur &gt;&gt;</b>	
<b>Fonctionnalités de GenTree</b>	
<b>Afficher l'individu résultant</b>	/2
<b>Savoir le nombre de générations qui le sépare de la population initiale</b>	
<b>Afficher les caractéristiques de tous les ancêtres d'un niveau donnés</b>	
<b>Lisibilité et documentation</b>	/-0.5
<b>Respect des consignes</b>	/-0.5
<b>Total</b>	<b>/10</b>