

ULCO - L2 Mathématiques
Examen Algorithmique - 2h - Session 2
*Documents de cours autorisés - Pas de calculatrice - Pas de
livre*

7 juin 2017

Exercice 1 : Listes chaînées

On va décrire dans cet exercice une liste de sculptures qui seront chacune composée de plusieurs composantes.

Une composante sera décrite par 4 caractéristiques : un nom qu'on lui donne, **nom** de type chaîne de caractères, sa longueur **lo**, sa largeur **la** et sa hauteur **ha** tous 3 de type nombre réel (les nombres représenteront des mesures en cm)

Une sculpture (par exemple, "Les bourgeois de Calais" de Rodin) pourra être décrite par un tableau de composantes (au maximum 12 composantes). On mémorisera également le titre de la sculpture et le nombre de composantes modélisant cette oeuvre.

Exemple : Pour modéliser "les bourgeois de Calais", on pourra utiliser 7 composantes : une pour chacun des 6 personnages, et une pour le socle.

On considèrera une liste chaînée de **sculpture** où chaque sculpture sera modélisée par une structure comme énoncé ci-dessus. Dans cet exercice,

1. vous écrirez la structure "composante", puis la structure "sculpture" comme décrites ci-dessus. La structure "sculpture" contiendra bien entendu en plus des différents champs énoncés, un pointeur sur l'élément (sculpture) suivant s'il existe, ce pointeur vaudra "Null" si pas d'élément suivant.
2. vous écrirez une fonction "nbComposantes" retournant le nombre total de composantes utilisées dans toutes les sculptures de la liste ;
3. vous écrirez une fonction "nbHauteurUnMetre" retournant le nombre total de composantes utilisées dans toutes les sculptures de la liste qui ont une hauteur supérieure ou égale à 1 mètre ;
4. vous écrirez une fonction "trierComposantesSculpture" qui trie les composantes dans l'ordre croissant des volumes ($lo*la*ha$) de la sculpture dont le nom "Nom" est passé en paramètre. Pour le tri vous utiliserez le tri fusion.
5. Supposons maintenant que les éléments de la liste sont triés dans l'ordre alphabétique des noms des sculptures, écrirez une fonction "insérerSculpture" qui permettra d'insérer la sculpture "S" passée en paramètre, à sa place (en fonction de son nom et en respectant l'ordre alphabétique des oeuvres), dans la liste dont la racine est également passée en paramètre.

Exercice 2 : Arbres binaires

Nous allons considérer un arbre binaire dont chaque noeud contient une valeur entière qui lui est propre "val", et la somme des valeurs des noeuds composant son sous-arbre "sommeSA" (on ne comptera donc pas la valeur du noeud lui-même).

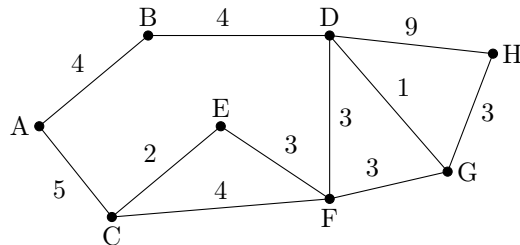
Dans cet exercice,

1. vous écrirez la structure d'un noeud comme décrit ci-dessus ;
2. vous écrirez une fonction "initialiser" qui prend en paramètres la racine de l'arbre et la hauteur de l'arbre et qui permet de construire un arbre binaire plein :

- la valeur de chaque noeud sera demandée à l'utilisateur ;
 - la somme du sous-arbre de chaque noeud sera initialisée à 0.
3. vous écrirez une fonction "calculSousArbre" qui prend en paramètre un noeud, qui permet de mettre à jour la variable "sommeSA" de ce noeud.

Exercice 3 : Graphes

On considère le graphe G suivant :



1. Le graphe G possède-t-il une chaîne eulérienne ? Si oui en donnez une, si non justifier.
2. Le graphe G possède-t-il un cycle eulérien ? Si oui en donnez un, si non justifier.
3. A l'aide de l'algorithme de Dijkstra, déterminez le plus court chemin allant de A à H . Vous donnerez les différentes étapes de l'algorithme.