Univérsité Hassan II Casablanca

Ecole normale supérieure (ENS) Casablanca

Module : structures de données et langage C

Parcours 2ème année LEESM **Pr M. Lahby**

Min Projet : Gestion de transport

28 Avril 2021

Objectif

L'objectif de ce projet est de développer une application en langage C, permettant d'assurer la gestion d'un service de transport de personnes sur un trajet donné.

Description

Une entreprise offre un service de transport de personnes dont elle désire automatiser la gestion.
Elle possède un parc de véhicules et emploie des chauffeurs.
Un véhicule est identifié par un numéro et caractérisé par sa marque, sa date d'achat, ainsi que le nombre de passagers qu'il peut transporter. Les véhicules sont répartis en différentes classes (voitures de tourisme, mini-bis, etc...). Les chauffeurs de l'entreprise sont identifiés par un numéro (leur matricule), et caractérisés par leur adresse et leur numéro de téléphone. Un chauffeur donné est autorisé à conduire une ou plusieurs classes de véhicules.
Un trajet est identifié par un code et caractérisé par une ville de départ, une ville d'arrivée et une distance (en km). Un transport est effectué à une certaine date sur un certain trajet. Lors d'un transport, une voiture est conduite par un chauffeur et transporte un certain nombre de personnes. Une voiture ne peut être sollicitée qu'une fois dans une même journée pour un trajet.

Travail à faire :

On devra pouvoir :

- ✓ Mémoriser et sauvegarder l'ensemble des données sur fichier ;
- ✓ **Pour les véhicules** :
  - Afficher tous les véhicules, les véhicules disponibles à une date donnée,
  - Ajouter un nouveau véhicule / mise à jour ses informations
- ✓ **Pour les chauffeurs :**
  - Afficher la liste de tous les chauffeurs, les chauffeurs disponibles à une date donnée
  - Ajouter un nouveau chauffeur / mise à jour ses informations
  - supprimer un chauffeur
- ✓ **Pour les transports :**
  - Programmer un transport en affectant un chauffeur à un véhicule pour une date et un trajet donnés
  - Pour une date donnée, lister les transports programmés (matricule véhicule, nom chauffeur, informations du trajet)

– Remarque : par défition, si un même mot apparaît plusieurs fois au sein d'une même phrase, on ne stocke qu'une seule fois l'index correspondant ;

- ✓ find qui permet de retrouver tous les indices d'occurrence d'un mot (par ordre croissant) ;
- ✓ findCooccurrences qui permet de retrouver tous les indices de cooccurrence d'un groupe de mots (par ordre croissant) ;
- ✓ isBalanced qui détermine si l'arbre est équilibré ou non. Un arbre est équilibré si pour chacun de ses noeuds, la hauteur du sous-arbre de droite de ce noeud ne diffère pas de la hauteur du sous-arbre de gauche de plus d'une unité ;
- ✓ getHeight qui renvoie la hauteur de l'arbre ;
- ✓ getAvgDepth qui renvoie la profondeur moyenne de l'arbre (i.e. la moyenne de la profondeur de chaque noeud de l'arbre).

Il n'est pas nécessaire d'implémenter la suppression de clés pour ce problème. En ce qui concerne la fonction getTotalNumString, si un même mot apparaît plusieurs fois dans une même phrase, il ne doit compter que pour un. Par exemple, l'insertion mot par mot de la phrase :
"foo bar foo baz"
dans un arbre vide conduit à une seule insertion de foo (et de bar et baz). La méthode getTotalNumString sur ce même arbre renverra 3

1.3 Fichiers fournis

Nous vous fournissons les fichiers suivants :

- ✓ **time_machine.txt** contient la version primaire du livre The Time Machine telle que fournie par le projet Gutenberg ;
- ✓ **time_machine_formated.txt** contient une version formatée du livre. C'est avec ce fichier là qu'on va travailler ;
- ✓ **SequentialSet.h** contient l'interface de la structure du même nom ;
- ✓ **BinarySearchTree.h** contient l'interface de la structure du même nom ;
- ✓ **main.c** contient un programme de test.

2 Rapport et analyse théorique

Il vous est demandé de fournir un rapport épousant le canevas suivant :

1. Ensemble séquentiel :
   - (a) Décrivez et justifiez votre implémentation de l'ensemble séquentiel.
   - (b) Expliquez le fonctionnement de la fonction intersect.
   - (c) Identifiez le pire cas et sa complexité pour :
     - i. l'opération d'insertion d'une position (en fonction de n, le nombre d'éléments dans l'ensemble).
     - ii. l'opération de test de présence d'une position (en fonction de n, le nombre d'éléments dans l'ensemble).
     - iii. l'opération d'intersection de deux ensembles (en fonction de m, le nombre total d'éléments dans les deux ensembles).
2. Arbre binaire de recherche :
   - (a) Décrivez vos choix d'implémentation pour l'arbre binaire de recherche.
   - (b) Pour les deux fonctions getAvgDepth et isBalanced :

i. Donnez le pseudo-code en vous basant sur l'implementation d'arbre binaire utilisée dans le cours théorique.

ii. Etudiez leur complexité dans le meilleur et le pire cas.

(c) Tracez l'évolution de la hauteur et de la profondeur moyenne des noeuds de l'arbre en fonction du nombre de clés qu'il contient. Commentez ces deux graphes. Pour recueillir les informations nécessaires, vous pouvez modifier le fichier main.c.

3. Complexité de FindCooccurrences

(a) En supposant que vous avez implémenté une insertion dans l'arbre binaire sans équilibrage, étudiez la complexité de la fonction FindCooccurrences dans le cas d'une recherche de deux mots en fonction du nombre de mots n dans l'arbre et en supposant qu'un mot apparaît au plus dans k phrases différentes. Expliquez le meilleur et le pire cas.

(b) Que deviennent ces complexités si on suppose que l'arbre est équilibré ?

3 Deadline et soumission

Le projet est à réaliser **individuellement** pour le 20 Juin 2021 à **23h59** au plus tard. Le projet est à envoyer par e-mail à l'adresse suivante : mlahby@gmail.com. Il doit être rendu sous la forme d'un dossier compressé .zip contenant :

(a) Votre rapport (4 pages maximum) au format PDF. Soyez bref mais précis et respectez bien la numération des (sous-)questions.

(b) Un fichier BinarySearchTree.c contenant l'implémentation de la structure correspondante.

(c) Un fichier SequentialSet.c contenant l'implémentation de la structure correspondante.

Respectez bien les extensions de fichiers ainsi que leur nom pour les fichier *.c (en ce compris la casse). Seule la dernière version soumise sera prise en compte.
Ceci implique que :

- Un projet non rendu à temps recevra une cote nulle
- En cas de plagiat avéré, l'étudiant se verra affecter une cote nulle.
- La note de ce projet représente 20% de la note du module