CSI2531, Hiver 2001
Devoir 2
A remettre, Vendredi 16 Fevrier, 11h00

Valeur: 7.5% Noté sur 100

1 Problème Ecrit 1: Compression par codage Huffman 15 points

1. Veuillez construire un arbre Huffman pour les paires de lettre/probabilité suivantes: (e, .45), (h, .13), (i, .12), (l, .16), (o, .09), (p, .05).
   S'il y a égalite dans les probabilités, veuillez décider d'une stratégie pour résoudre cette égalité (et vous y tenir). Veuillez utiliser la convention suivante: pour chaque noeud dans l'arbre, l'enfant de gauche a une probabilité plus basse que celui de droite.

   Veuillez labeller les branches de votre arbre de facon a ce que chaque branche de gauche recoive la valeur "0" et chaque branche de droite recoive la valeur "1".

2. En utilisant votre arbre Huffman construit dans la partie précédente, veuillez décoder le message suivant: 101100.

3. En utilisant votre arbre Huffman construit dans la partie 1, veuillez encoder le message: "help".

4. Veuillez calculer le nombre moyen de bits par caractère étant donné le codage des lettres trouvé dans la partie 1 et leur probabilités. Montrez votre formule de calcul.

2 Problème Ecrit 2: Compression par codage Lempel Ziv 15 points

1. Veuillez construire un codage Lempel-Ziv pour la chaîne de caractère suivante: "aabbcabcaaabcabcabcbaabcaa". Veuillez montrer l'arbre résultant de votre codage.

2. Combien de bits sont nécéssaires pour coder ce message? Quel montant d'éspace est donc sauvergardé par votre codage (en nombre de bits ou d'octets)?

3. Veuillez décoder le message suivant: "0a0b1b3b0c4b".

3 Problème de Programmation 1: Dé-compression de code Huffman 35 points

Un message a été placé sur le site Web pour vous. Neanmoins, afin de ne pas utiliser trop d'éspace sur le disque et trop de temps pendant la transmission, le message a été encodé par la méthode de codage Huffman et le fichier a été sauvegardé en format binaire.

Vous allez recevoir un programme contenant les méthodes nécéssaires pour construire un arbre Huffman basé sur la probabilité de chaque lettre de l'alphabet en Anglais (pour simplifier le problème, seulement les lettres capitales et le caractère "espace" sont permis dans le message original). Votre tâche consiste a écrire la partie du programme qui vous permettra de décoder le message basé sur cet arbre Huffman.

Plus spécifiquement, votre programme va:

1. Lire le fichier binaire `encoded.bin` contenant le message codé.

2. Traverser l'arbre Huffman (en décodant le message) guidé par le contenu de `encoded.bin`. **Note:** Par convention, chaque branche de gauche correspond a "0" et chaque branche de droite correspond a "1".

3. Ecrire le message décodé dans le fichier `decoded.txt`

Vous devez completer le programme inclus sur le site Web. De plus amples details sont donnés dans le fichier `README.txt`

CONCOURS DE PROGRAMMATION: Prix et Règles du Concours

Le premier etudiant parvenant a décoder le message codé en utilisant le programme de décodage Huffman écrit pour le problème précédent gagnera le livre suivant en prix:

`Lippman and Lajoie, ''C++ Primer'', Addison-Wesley, 1231 pages.`

Nous remercions Addison-Wesley pour leur don dans le contexte de ce concours!

Règles de Participation du Concours:

1. Veuillez vérifier la page Web afin de voir si quelqu'un a déjà remporté le prix. Dans ce cas, le concours est fini. Si personne n'a encore remporté le concours, alors faites les choses suivantes:

2. Envoyez un e-mail à:
   `nat@site.uottawa.ca` ET `lucia@site.uottawa.ca` avec votre nom, numéro d'étudiant, le message décodé et tous les fichiers (en attachement) nécéssaires à l'execution de votre programme.

3. Le vainqueur sera l'étudiant dont l'e-mail atteindra nos boites aux lettres éléctroniques le premier, et qui contiendra un programme correct et un message décodé également correcte.

4 Problème de Programmation 2: Addressage Direct & Recherche Binaire 35 points

Dans ce programme vous allez modifier le fichier trié généré dans le devoir 1 (le fichier avec numéro d'étudiants, mots de passe et noms). Votre programme doit iterer une boucle demandant à l'usager les informations suivantes:

```
num\'ero d'\'etudiant :
ancien mot de passe:
nouveau mot de passe:
```

```
confirmez le nouveau mot de passe:
Continuez ou Sortez du programme (C/S):
```

Pour ce devoir, nous allons prétendre que le fichier d'entrée est trop grand pour tenir en mémoire principale. Vous devez utiliser une rechèrche binaire (la clé est le numéro d'étudiant) afin de trouver l'enregistrement dont le mot de passe doit être changé. Dans votre rechèrche binaire, vous examinerez chaque clé en accèdant diréctement à la position du fichier à laquelle l'enregistrement correspondant est sauvegardé; Une fois la position trouvée, l'enregistrement doit être modifié en changeant le mot de passe.

Veuillez utiliser les méthodes `seekg` et `seekp` afin de vous déplacer vers la position appropriée du fichier.

Une solution pour le premier devoir est postee sur la page Web. Cette solution peut etre utilisee comme la base de votre nouveau programme. (Mais vous pouvez egalement utiliser le code que vous avez vous-meme ecrit dans le devoir précédent).

Tous les fichiers et toutes les informations nécéssaires a ce devoir se trouvent à l'addresse: http://www.site.uottawa.ca/ nat/Courses/GF-Cours/Devoir2/