Université Bordeaux I
U.F.R. Mathématiques et Informatique

Master Bio-Informatique
Année 2003-2004

Algorithmique et Structures de Données
Projet "Codage de Huffman"

Alain Griffault

12 février 2004

Avertissement

L'objectif du projet est le développement de classes Java pour la compression de données. Les codages de HUFFMAN procèdent ainsi : à partir d'un texte initial, un code est créer ; puis ce code peut servir pour coder et décoder des textes. Le code créer est *optimal* pour le texte initial, mais il est applicable à tous les textes utilisant le même alphabet.

Travail à réaliser

Signer vos fichiers

Question 1 *Vous devez ajouter vos noms et prenoms aux endroits prévus dans les fichiers :* `ArbreBinaire.java`, `CodageHuffman.java`, `Info.java` *et* `Tas.java`.

Compréhension des sources données

Question 2 *Vous devez ajouter des commentaires aux endroits prévus dans les fichiers :*

ArbreBinaire.java *aux méthodes :*
- `void setCodage(String st)`
- `String getCodage(String st)`
- `ArbreBinaire getArbreBinaire(char c)`
- `String toString()`

CodageHuffman.java *au constructeur et aux méthodes :*
- `CodageHuffman(String st)`
- `ArrayList getAlphabet(String st)`
- `String getCodage(String st)`
- `String getDecodage(String st)`
- `String toString()`

Tas.java *au constructeur et aux méthodes :*
- `Tas(Collection c)`
- `int getComparator(int i)`
- `void entasser(int i)`
- `void inserer(Object o)`
- `ArbreBinaire extraireMinimum()`

Implémentation du codage de Huffman

L'algorithme construit du bas vers le haut l'arbre A, correspondant au codage optimal appelé **codage de Huffman**. Il commence avec un ensemble C de feuilles et effectue une série de $|C| - 1$ *fusions* pour créer l'arbre final.

L'ensemble C de feuilles correspond à l'ensemble des couples (caractère, nombre d'occurrences) pour chacun des caractères trouvés dans le texte.

L'ensemble C est transformé en fonction des nombres d'occurrences en une structure de **Tas**.

Ce tas sera utilisé comme une file de priorité, et étendu en fusionnant les couples ayant les fréquences les plus faibles.

Plus précisement :

```
Huffman(texte) {
  C[c] <- nbOccurrences(c,texte);
  tas  <- creerTas(C);
  entier n <- cardinal(C);
  pour i de 1 à n-1 faire {
    aux <- creer ArbreBinaire;
    aux.fg <- extraireMinimum(tas);
    aux.fd <- extraireMinimum(tas);
    aux.info <- nbOccurrences(aux.fg)+nbOccurrences(aux.fd);
    inserer(aux, tas);
  }
  enregisterCodage(aux);
  retourner aux;
}
```

Question 3 *Vous devez écrire le code de la méthode* `ArbreBinaire creerCodage(Tas tas)` *et son commentaire dans le fichier* **CodageHuffman.java**.

Implémentation du décodage de Huffman

Le codage est utilisé pour coder un texte, mais il est aussi utilisé pour décoder un texte codé.

Question 4 *Vous devez écrire le code de la méthode* `String getDecodage(String st)` *et son commentaire dans le fichier* **ArbreBinaire.java**

Remarques diverses

Essayer de mettre vos commentaires de façon qu'ils soient traités correctement par l'outil javadoc. Essayer pour cela la commande :
`javadoc -private -author *java`
et regarder le fichier `index.html` généré.

L'algorithme donné est correct lorsque le texte initial contient au moins deux caractères différents. Dans le cas ou le texte ne contient qu'un seul caractère, vous pouvez au choix :
- ne rien faire,
- ne pas générer de code, et avertir l'utilisateur,
- adapter l'algorithme.

Contraintes

Récupération du sujet

sujet	http ://dept-info.labri.fr/ griffaul/ASD/Projet/sujet.ps
source	http ://dept-info.labri.fr/ griffaul/ASD/Projet/*.java
classes	http ://dept-info.labri.fr/ griffaul/ASD/Projet/*.class

Les classes correspondent à une correction possible. Vous pouvez les tester avec par exemple la commande :
`java CodageHuffman ‘‘gjhgj gkhgsq’’`

Remise du travail

Date de remise du travail	Lundi 1 mars à 12 heures
Binôme accepté, trinôme refusé	
Le rapport comprend	Les quatre fichiers java un fichier texte éventuel en complément explicatif
Remise du rapport	e-mail à Alain.Griffault@labri.fr