Algorithmique et structures de données : Mission 5

Groupe 1.2: Ivan Ahad - Jérôme Bertaux - Rodolphe Cambier
Baptiste Degryse - Wojciech Grynczel - Charles Jaquet

28 novembre 2014

Rapport écrit par Rodolphe Cambier, Ivan Ahad

Introduction

La mission pour cette semaine était de créer un programme permettant de compresser et de décompresser des textes à l'aide du codage de Huffman.

Questions par rapport au programme

Question 7

IntputBytestream et OutputBytesream permettent respectivement la lecture et l'écriture en bit à bit. La méthode close est par exemple appelée dans une situation telle que notre texte à compresser contient seulement 11 fois le caractère "a". Ainsi, la compression se fera sur 11 bits. La méthode "close" va donc combler les 5 bits manquant pour atteindre le multiple de 8. La conséquence sur notre programme est que l'on rajoute un caractère "EOF" (end of file) pour ne pas tenir compte des 0 à la fin du fichier, ainsi la lecture s'arrêtera lorsque l'on tombe sur ce EOF.

Question 8 - Diagramme UML

La classe commune entre le programme de compression et de décompression est OutputByteStream.
La classe Entry et Node implémentent l'interface comparable.

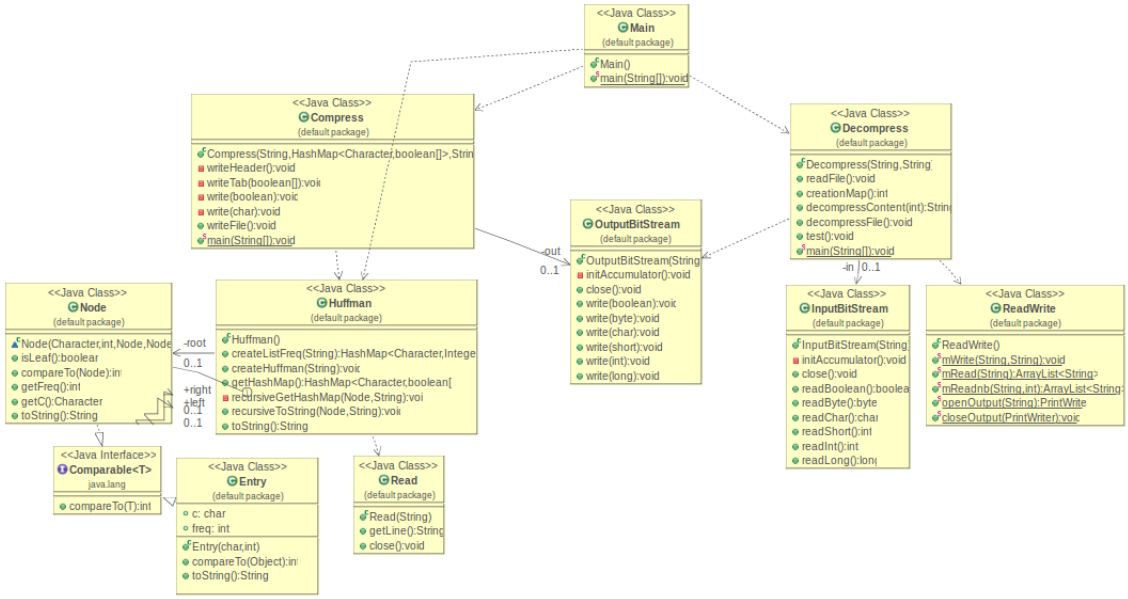
Question 9

Le taux de compression dépend à la fois de la taille du fichier et du nombre de caractères différents utilisés. Dans un cas extême, si de nombreux caractères différents sont utilisés dans un texte très court, le fichier compressé sera plus lourd que le fichier d'origine. C'est dû au fait qu'on ajoute le mapping des lettres à leur représentation en bits en début du fichier compressé.

Par exemple, tenter de compresser un fichier contenant seulement les caractères "abdefghijklmnopqrstuvwxyz", le fichier original est de 26 bytes, et le fichier compressé est de 110 bytes.
Le programme ne devient efficace qu'à partir d'un certain nombre de caractères.

Question 10

Lors de la compression, les retours à la ligne ne sont pas pris en compte. Ainsi, lorsque le texte est décompressé, celui-ci se retrouve sur une seule ligne. A part cet inconvénient, la compression et la décompression sont sans perte.



Pour tester cela nous avons utilisé un texte comportant divers types de caractères.