Team ID: 22

Topic: Huffman Compression

Description:

Huffman Compression is a very useful compression algorithm that is utilized in several commonly used compression schemes. For more information on the details of Huffman Compression, see: http://en.wikipedia.org/wiki/Huffman_coding. An example will also be provided in the instructions below.

In This assignment, you will read in the representation of a Huffman tree from a text file, and either encrypt (characters to Huffman codes) or decrypt (Huffman codes to characters) the rest of the data from the test file.

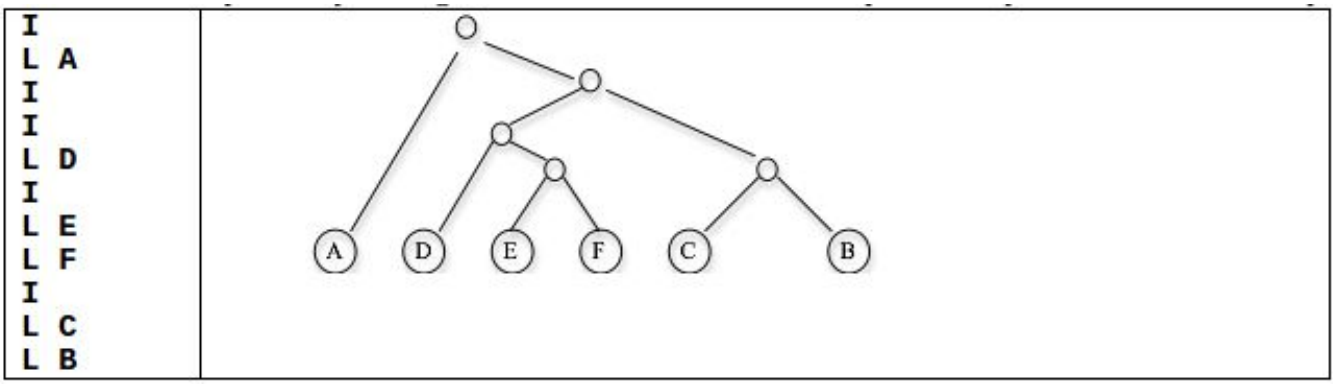
Huffman tree example:

The input file will consist of a number of lines of text. Each line will be one of two possibilities:

1) The single letter "I". This indicates that the node is an Interior node in the Huffman tree. For simplicity, in this case we will consider the root to also be an interior node. Due to the nature of Huffman trees, all interior nodes will have two children.

2) The single letter "L" followed by a single space, followed by a single letter. This indicates a Leaf node in the Huffman tree. The second letter on each line can be any letter, number, or special character(!,@,#,$....etc), but each character can appear in at most one line (No repetition). These second letters represent the characters that can be encoded by the Huffman tree. For simplicity, assume that any subset of the alphabet in the file will always be a contiguous sequence starting at 'A'. For example, if the file contains only 5 letters, the leaf nodes will be 'A', 'B', 'C', 'D' and 'E'. However, they do not have to appear in the file in any particular order.

Below is a simple example of the tree building algorithm:



The file in the left box represents the Huffman tree shown in the right box. To obtain actual Huffman codes from this tree, assign the value '0' to each left edge and the value '1' to each right edge. Thus,

for example, the character 'D' would have the Huffman code "100" and the character 'F' would have the Huffman code "1011".

The resulting table for your reference.

Index	Letter	Huffman Code
0	A	0
1	B	111
2	C	110
3	D	100
4	E	1010
5	F	1011

I/O Format:

The test file will follow this format.

Tree information

Operator

Length

Data

Tree information is the same format as described above, in the form of I or L+character. At the end of the tree, the character E will be added to signify the end of building the tree. (To make it easier for you to break)

Operator will be either 0 or 1. Zero means encryption (Character to Huffman code), while One means decryption(Character to Huffman code).

Length is an indicator of how many elements are in the data set. *Will be under 1000

Data will either be a sequence of characters (if the operator is 0) or a sequence of huffman codes (if the operator is 1). Characters are within ASCII.

Using the above tree as an example,

Input1.txt

I
L A
I
I
L D
I
L E
L F
I
L C
L B (up to here, tree information) (parenthesis are notes, not part of the text file)
E (end of tree)
0 (encode)
1 (1 character)

A (data to be processed, in this case encryption)
Output1.txt

0

Input2.txt

I
L A
I
I
L D
I
L E
L F
I
L C
L B
E
1
3
111
0
100
Output2.txt

B

A

D

Time limit: 25s

Memory limit: 0.5mb