

BBM203

ASSIGNMENT 4

I AM TAKING THIS COURSE FOR THE LAST TIME
GOODBYE :)



HUFFMAN ENCODING

2021

ÖZGE KÖKYAY

21727484

Encoding Algorithm

1. Read input file and create frequency table for each character and their frequencies.
2. In function createQueue, read the characters and their frequencies and create HuffmanNode for each character. Add nodes to the priority queue. The character that has lowest frequency has the highest priority in the queue.
3. If queue has more than 1 element create one internal node with these 2 nodes that has highest priority in the queue and pop queue 2 times. Frequency of the internal node is equal to sum of 2 nodes frequencies. Push this node to the queue.
4. Last element in the queue is the root node of the tree.
5. In function encodeTree, until node is null traverse the tree and add 0 when go to the left and add 1 when go to the right to the encoding, create a map for character and corresponding encoding of the character.
6. Print encoding of the text and delete all nodes in the tree.

Decoding Algorithm

1. Encode the frequency table with functions that used while encoding.
 - createQueue
 - encodeTree
2. Read file and create a temporary HuffmanNode* curr for traversal in the tree and when the character is 0 curr becomes curr->left and when the character is 1 curr becomes curr->right until reach a leaf node.
3. When reach a leaf node print it.

List Tree Command

I listed the tree as in the image, used `*--` to represent the parent nodes, branches and siblings. As you go to up in the tree, it means you add 1 to encoding, as you go to down in the tree, it means you add 0 to encoding. Here is an example tree of encoding the text "go go gophers".

