HACETTEPE ÜNİVERSİTESİ

BBM203: SOFTWARE LABORATORY I

Fall 2020

Programming Assignment 4 Report

Emirhan Topcu 21827899

Subject: Trees

Programming Language: C++

Encoding Algorithm

1. Program reads the file given command line, creates a vector of Node pointers and puts every single character with their frequencies in this vector.
    1.1. Reads every char in the input file and counts them. if the char is not in the frequency list creates a node pointer
    1.2. If the char exist in the frequency list, it increases that char's Node's frequency by 1.
    1.3. Sorts the list and returns it
2. Makes a tree and returns the pointer of the root node.
    2.1. Takes the first two element from the sorted frequency list.
    2.2. Creates a new node with sum of the two nodes' frequencies as frequency and these two nodes as child nodes.
    2.3. Puts the new node to the frequency list and sorts it again.
    2.4. If there’s only one element left in the frequency list return it as root node pointer.
3. Creates a map for storing the codes of each character.
4. Traverses the tree and gets the code table.
    4.1. Checks the given node if its a leaf node, if not calls the same function again for the right direction.
    4.2. Each time it calls itself the code gets extended with zero or one.
5. Encodes the input line by using the code table.
    5.1. Takes chars in the input line one by one and updates the encoded string line with their codes.
6. Prints the encoded message.
7. Creates a file to store the tree for next uses. (serializing)
    7.1. Checks the node if its null, if not calls the same function for left and right nodes
    7.2. Returns called functions and its label.

Decoding Algorithm

1. Reads the serialized data file and deserializes it
    1.1. Uses a queue to store the chars, then sends it to the extension function.
        1.1.1. For every char in the queue, it checks them if they represent a null node pointer, if not creates a node with the char.
        1.1.2. For the child nodes, calls the function again.
        1.1.3. Returns root node finally.
2. Reads the encoded line from the input file.
3. Runs a while loop until all the bits in the encoded line are read, calls decode function for every code.
    3.1. Checks if the node is a leaf. If so, prints node's label.
    3.2. Increases the index for next operation.
    3.3. Calls the function again according to the next bit in the encoded line.

Listing Algorithm

1. For current node, it looks if the node is a leaf, if so prints the label - if the label is not a letter, it prints " -* " -.
2. Prints some characters according to the depth count to show the depth of the nodes on screen.
3. If the node is not a leaf calls the same function for child nodes.