ITU Computer Engineering Department
BLG 223E Data Structures, Fall 2021
Recitation #4
Due December 07, 2020 11:59pm

"The struggle you're in today
is preparing the strength
you need tomorrow.."

Problem Defintion

In this recitation, you have to implement Huffman coding, an entropy coding algorithm used in computer science for data compression. In 1951, David A. Huffman suggested it as a term paper of information theory lecture at Massachusetts Institute of Technology (MIT). It relies the idea of using frequency-sorted binary tree. Huffman's method creates a unique code which consists of 1s and 0s for each symbol (or character). It produces shorter code for frequently used ones and longer code for less frequently used ones. The Huffman method constructs a binary tree based on the frequency of usage of the characters (or symbols) in the data. Each symbol is represented on the binary tree by the route that runs from the root to the leaf node of that symbol. The left branch is depicted by '**0**', whereas the right branch is depicted by '**1**'. Figure 1 visualizes Huffman coding algorithm.

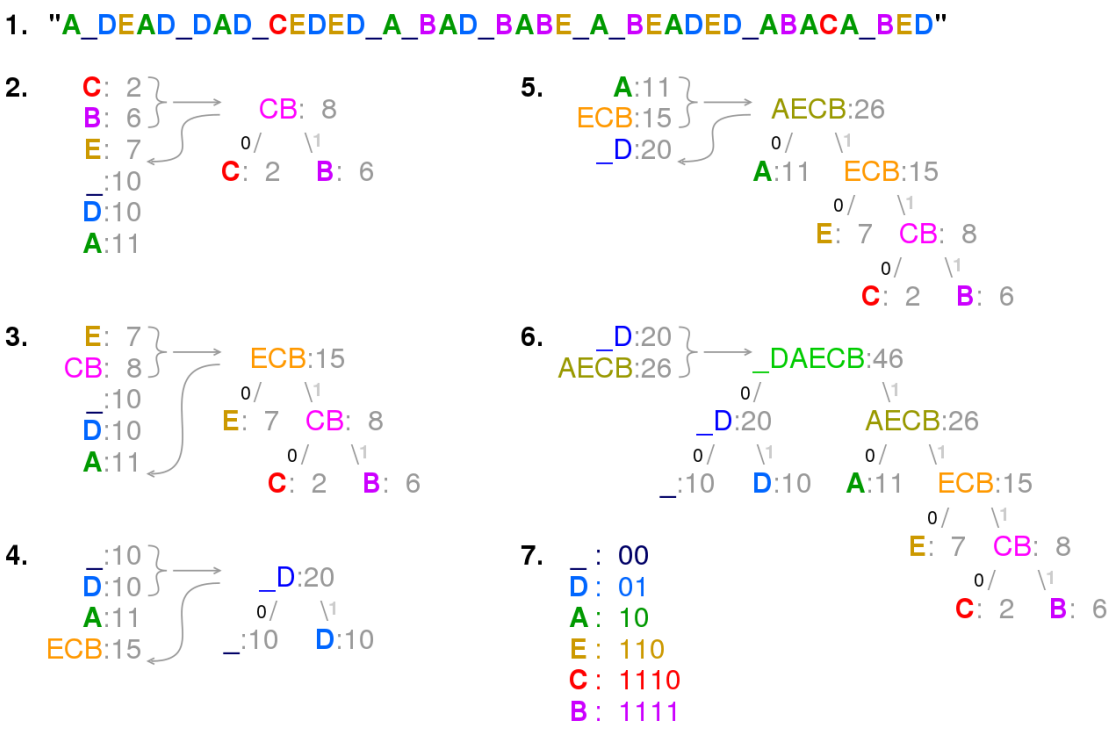

Figure 1: Huffman coding visualisation

Compression ratio is calculated as given in Equation (1). The Huffman algorithm is an efficient technique for compression of data that has few character variants. However, the generated tree must be appended to the compressed data. This requirement reduces the compression efficiency.

Compression Ratio = bits required before compression / bits required after compression. (1)

Implementation

The necessary libraries, some of functions and structs for the solution are be given in code. You have to recall arrays, linkedlist, recursive functions and binary tree subjects for this recitation. You have to follow up below steps to implement this recitation;

1. Read input text from file. Find frequencies for each symbol.
2. Create a leaf node for each symbol and add it to the priority queue.
3. While there is more than one node in the queue:
   (a) Remove the two nodes of highest priority (lowest frequency) from the queue
   (b) Create a new internal node with these two nodes and with probability equal to the sum of the two nodes' probabilities. Place the new node as parent of its components in the tree.
   (c) Add the new node to the queue.
4. The remaining node is the root node and the tree is complete.
5. Use this tree to produce code for each symbol.
6. Calculate the compression ratio.

Submission Rules

- Make sure you write your name and number in all of the files of your project, in the following format:
  /* @Author
  Student Name: <student_name>
  Student ID : <student_id>
  Date: <date> */
- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not **share or copy code** from your classmates or from the Internet. You should submit your **own, individual** homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- Use comments wherever necessary in your code to explain what you did.
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**