Project Name: Text File Compressor

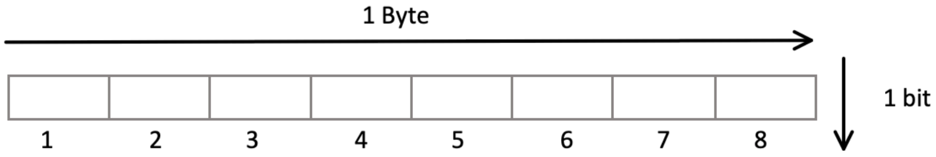
Objective: I have read DSA and solved the questions as many as possible. In companies, I know that I have to Implement my knowledge and experiences on a project. So I worked on a project. In this project, my actual motive is to make text files more portable so I have made a "text file compressor" using the knowledge of DSA.

Technology: C++

Platform: vim editor using terminal

Approach: In this project I don’t want to loose any data so I have used “Huffman Coding” because “Huffman Coding” is a lossless data compression algorithm ie, it is a way of compressing data without losing data of the information.

In “Huffman Coding” our main goal is to reduce the consumption of bits ie, we know that, every character takes 8 bits and we have to reduce the consumption of bits for every character, then only we can archive the compression. So in “Huffman Coding” we make a unique code of 0 and 1 for each character. Make sure that here 0 and 1 are in bit ie, we have to use each bit of a byte for storing the codes of the corresponding character.



Procedure:

There are mainly two major parts in Huffman Coding —

1. Build a Huffman Tree from input characters.
2. Traverse the Huffman Tree and assign codes to characters.

1. Build a Huffman Tree from input characters.

a. First of all, count frequency of characters in the text and then put them in a min-heap on the basis of the frequency of character by making nodes with character and count of their frequency.

b. Then extract two nodes with the minimum frequency from the min-heap.

c. Then create a new node with a frequency equal to the sum of the two node's frequencies. Make the first extracted node as its left child and the other extracted node as its right child. Add this node to the min-heap again.

Repeat this until the heap contains only one node. The remaining node is the root node and the tree is complete.

2. Traverse the Huffman Tree and assign codes to characters.

a. Calculate the path from the root to leaf node by adding 0 and 1 in the code when we go to the left and right respectively.

b. Then we should store the character and their code in Hashmap so that we can access the code and character of the corresponding character and code.

c. Now we have to replace the character with their code by making sure that the code is stored in bits, not in bytes.

Finally, we have achieved compression of a text file.

For decompression just travel on that bits and replace them with their corresponding character.

NOTE:- We have to give both compressed file and the file that contains code and their corresponding character