WHAT IS THE PROBLEM: The problem that was given to us was that we had to make a Huffman tree.

HUFFMAN TREE: Huffman tree is the way to compress the text that is stored in bits. Each of the character is stored in 8 bits character so if our file contain 8 letters the storing of that text will take 64 bits. But by using the Huffman tree we can store the bits by reducing the number of bits.

HUFFMAN TREE OPTIMIZED: The optimized Huffman tree use the frequency of the letter that is occurring in more in the file and then gives it the lowest code.

THE CODE: In the code we start by asking the user to give us the name or path of the file where the .txt is written.

- After this we asked the user whether he want to make optimal Huffman tree or normal tree.
- A very beautiful and attractive menu is displayed as you can see in the code
- In the normal tree we don't make use of the priority queue.
- The character is stored in the normal queue and one by one de-queued from the queue.
- In the optimal Huffman tree we use priority queue. The lowest the frequency the more the priority.
- We are using linked list based priority queue.
- We sort and extract character from queue until only one character remains.
- **CODING:** the coding is done by the in-order traversal of the tree. Each node of the tree contains a code string. As we move left the '0' is append in the code variable, and when we move right '1' is appended. That's how we getting coding for each character.
- **ENCODED:** The process of converting the normal text into code is called encoding. We take each letter from the original text and search for its corresponding node and then take its code and then append it into coded variable.
- **DECODING:** The process of converting coded text back normal form is called decoding. We read the code that is given to us character by character. If the character is '0' we move left and if the character is '1' we move right until we reach to the leaf node then we have our character.
- At the end we get ABR.

Result from the code

Original text aadbaaca:

Number of bits are equal to **64**.

Normal Huffman aadbaaca:

Number of bits of code are equal to **16.** Compression ratio is 4

Optimal Huffman aadbaaca:

Number of bits of code are equal to **13.** Compression ratio is 4.92