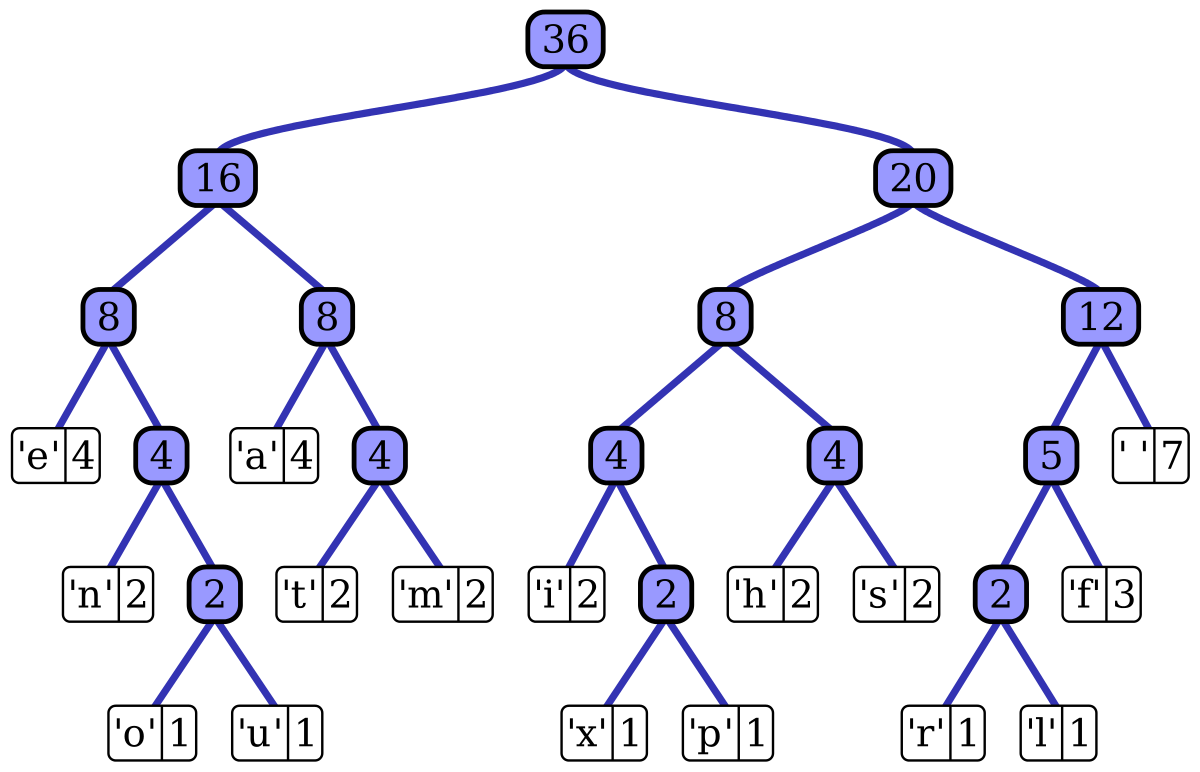
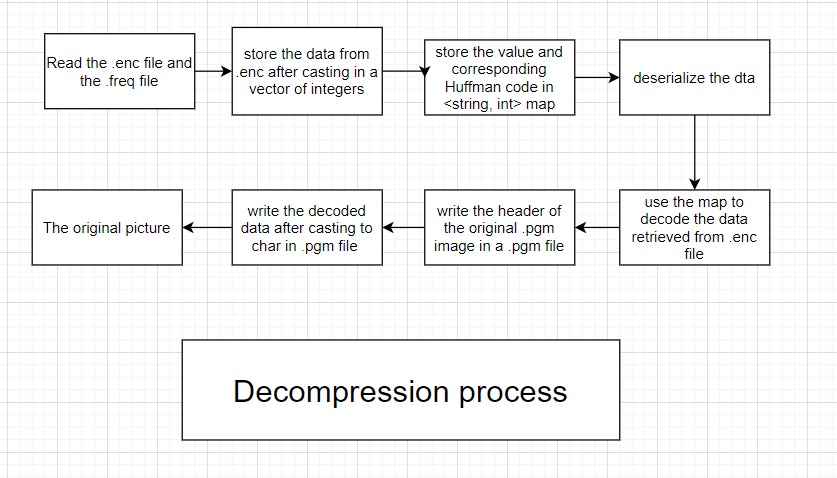
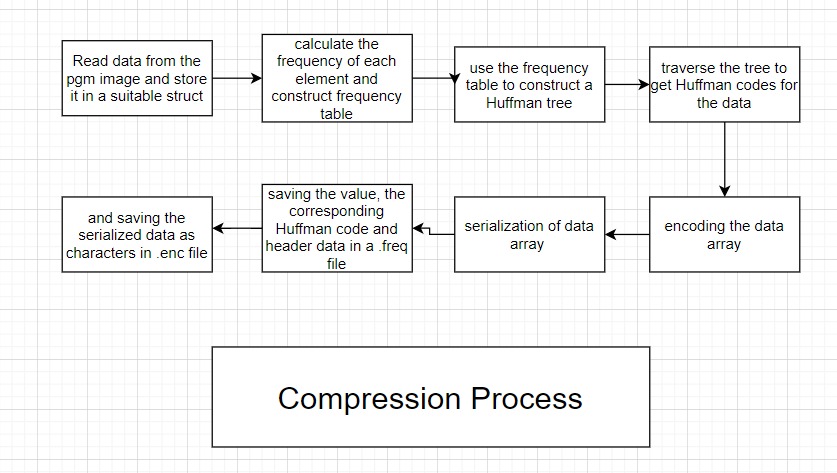
**Huffman Algorithm**

|  |  |  |
| --- | --- | --- |
| Ammar Al-Saeed Mohamed  9180914  Ammar.sa3ed00@gmail.com | Ahmed Sayed Ahmed  9180080  Ahmed.mohamed994@eng-st.cu.edu.eg | Mouaz Mohamed Hanafy  9181440  mouazhanafy120@gmail.com |
| Mohamed Khaled galloul  9181170  Mohamed.Attia99@eng-st.cu.edu.eg | Tarek Mohamed Rashad  9180691  Roka.2014.tr@gmail.com |  |

****

Huffman Tree

|  |  |
| --- | --- |
| * **Objective of the project:**   The objective of the program is to receive a portable grey map image (PGM) of type P5, read the pixel values and store them in a STD vector, then apply Huffman compression algorithm to compress the data. The program can also decompress the outputs of compression so that we return to the original size and format. The compression and decompression of the program are lossless.   * **Methodology of the code:**   We build a Huffman tree by creating a leaf node for each frequency of inputs and build amin heap by using these nodes and then we extract two nodes with the minimum frequency from the min heap. Create a new internal node with frequencies equal to the sum of the frequencies of the two nodes. Make the first node extracted as its left child, and the other node extracted as its right child and Add that node to min heap. We repeat these steps until one node remains  We generate the codes by traversing the Huffman tree starting from the root and while moving to the left child, write 0 to the array. While moving to the right child, write 1 to the array and nd ends when reaching a leaf node.   * **Results:**   Our program now can compress an image and save the data in two files (.enc and .freq). It can also read the two files resulting from compression to start decompression and return the original image without any data losses. | * **General info:**   Huffman Algorithm is an algorithm developed by David A. Huffman while he was a student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes". The main idea of Huffman code is to assign variable-length codes to input characters, and the code lengths are based on how frequently the input characters are repeated. The most frequent input gets the smallest code and vice versa. |

****

|  |  |
| --- | --- |
|  |  |

**Original File**

**The file after being compressed and decompressing the code file**