

**ALGORITHM AND PROBLEM SOLVING LAB**

**PROJECT REPORT**

IMAGE COMPRESSION USING HUFFMAN CODING

SUBMITTED BY:

Utkrisht Sharma – 16103116

Chaitanya Kumar – 16103244

Aarya Patel – 16103026

BATCH – B15

Problem Statement

Compressing 24bit .bmp image using Huffman Coding.

Relevance

The technique is best suited for lossless transfer of images.

Introduction

Huffman coding is a lossless data compression algorithm. Using this algorithm we can assign variable length bit codes to the data that is being compressed. The characters are assigned codes based on their frequency of occurrence and the characters with highest frequency are assigned smallest length bit codes and the characters with least frequency are assigned the longest length bit codes.

Every 24 bit .bmp file starts with a 54 byte header, the 18th byte specifies the width of the image and the 22nd byte specifies the height of the image. After the header the color of each pixel is specified by its RGB values. The color of the bottom right pixel is stored first. The colors are stored in the order of blue, green and red and each color takes 1 byte each. Therefore, each pixel requires 3 bytes.

Algorithm

1. Read .bmp image and store its width, height, size and information of pixels in imageInfo struct.
2. The information of each pixel is saved in an object of Pixel class containing information about the RGB values of the pixel, and the corresponding hex values.
3. No. of distinct colors are calculated and the frequency of each distinct color and its hex value is stored in an object of class color.
4. A Huffman tree is created using the information in step 3 and the bit codes are saved with their respective hex values and frequency in the color class.
5. The header of the new file is written
6. The entire Image Matrix is traversed and the corresponding bit code of the color of each pixel is written in the file.
7. The written file is saved with an extension of .cmpbmp.

Proposed File Format (.cmpbmp)

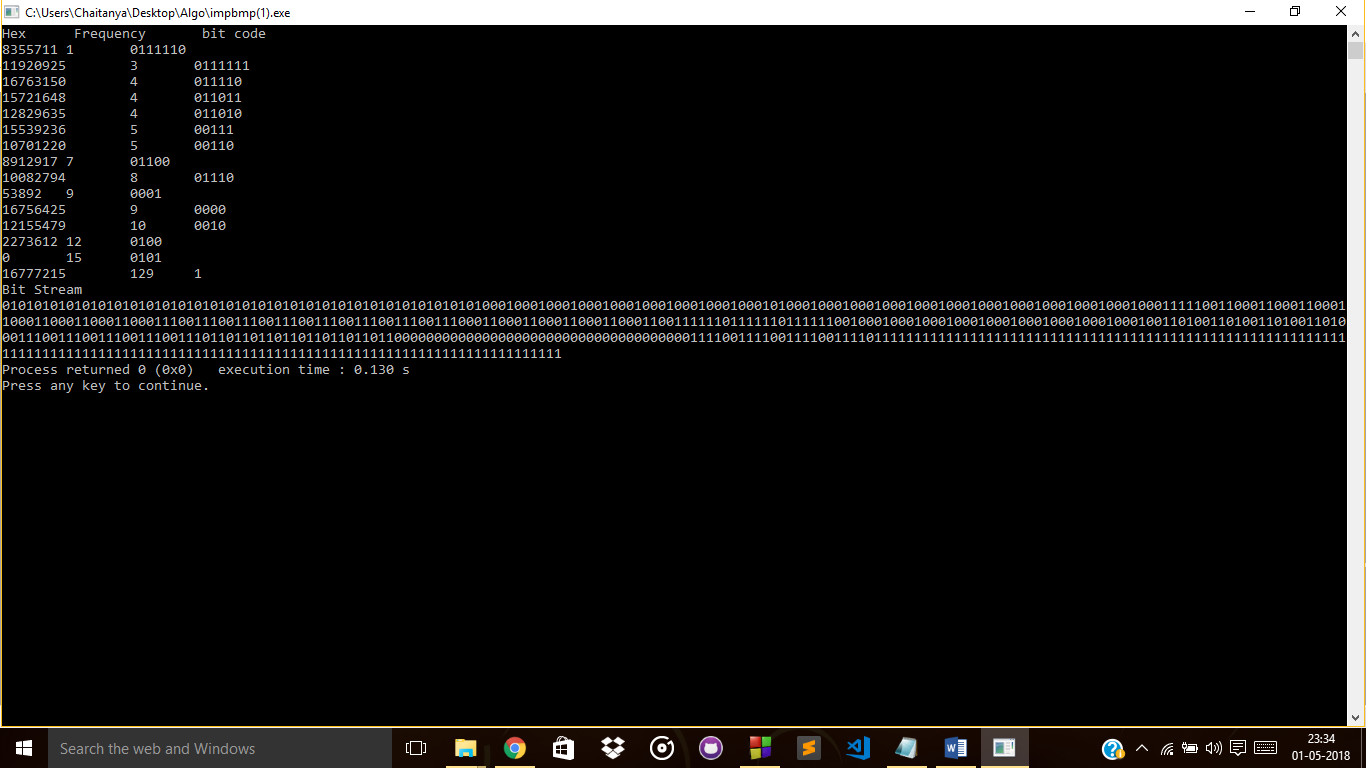
The first 4 bytes of the file contain the number of distinct colors in the compressed image. The information of hex value and its corresponding frequency is also stored for each distinct color where both hex value and frequency takes 4 bytes each.

The actual image contains the bit codes generated by Huffman coding and is the actual compressed data.

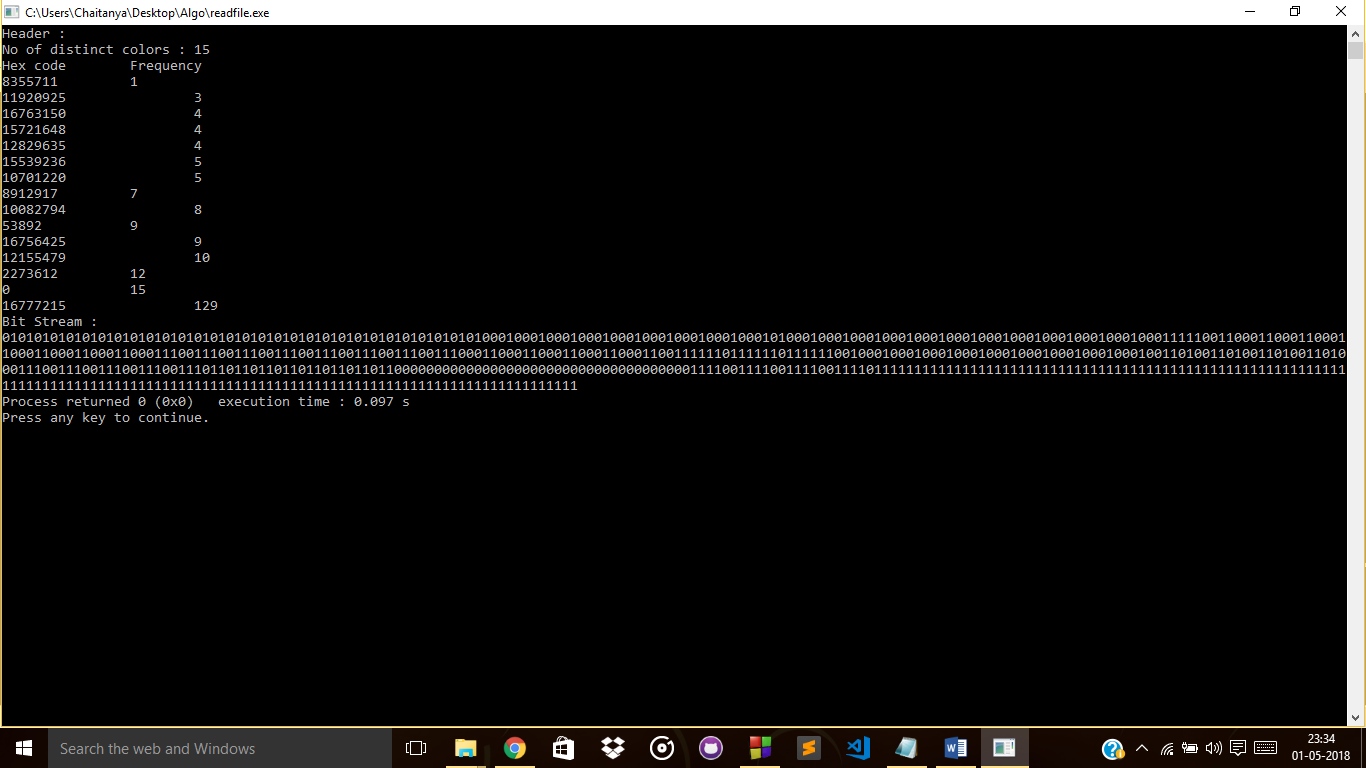
Using the information in header a Huffman tree is created and bit codes are generated which are used to decompress the image.

Screenshots

Input



Output



References

1. http://www.di.unito.it/~marcog/SM/BMPformat-Wiki.pdf
2. https://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding/
3. https://www.geeksforgeeks.org/unordered\_map-in-stl-and-its-applications/

Division of Work

Utkrisht – File reading and writing and debugging.

Chaitanya – Frequency calculation and distinct colors and debugging.

Aarya – Huffman Coding.