

# Huffman Encoding

[alaa.mera99@eng-st.cu.edu.eg](mailto:alaa.mera99@eng-st.cu.edu.eg)

[aya.okasha99@eng-st.cu.edu.eg](mailto:aya.okasha99@eng-st.cu.edu.eg)

[mariam.aziz99@eng-st.cu.edu.eg](mailto:mariam.aziz99@eng-st.cu.edu.eg)

[Meeran.elsayed99@eng-st.cu.edu.eg](mailto:Meeran.elsayed99@eng-st.cu.edu.eg)

[nouran.youssef99@eng-st.cu.edu.eg](mailto:nouran.youssef99@eng-st.cu.edu.eg)

## 1. Introduction

Nowdays , it is needed to deal with small files in size , but some files when be created has a big size so we need to compress them having same properties and information .huffman encoding is an algorism compress any file to be smaller in size (the basic concept of file compression )

The principle is using variable-length encoding instead of fixed-length encoding of character (0,1) depending on the frequency of each character in the text to control the size

The process work by create a binary tree of nodes , at first each node is a leaf node contain the character and its frequency. Internal nodes contain the frequency and 2 child nodes , as known '0' refer to the left child and '1' refer to right child .the latest tree has n leaf nodes and n-1 internal nodes

## 2. Motivation

We have chosen this project for some reasons

- After studying the idea of project , We made sure to apply the parts that are studied in this semester like struct and tree and use them into algorisms like Huffman
- Inquisitiveness and new knowledge about image process
- Inquisitiveness of compression as a whole

## 3. Resources

- Libraries

Iostream , String library , vector library , cstdint , queue , unordered\_map , bitset

- Tools  
std::vector , std::unordered map ,  
std:: string , std::ifstream ,  
std::ofstream

## 4. Challenges and problems

The problems that come are a challenge for us and we could address

- Image reading : It took us a while to connect to a simple method by reading the picture through streams Especially ifstream , and reading of text is different from binary files
- We have a connect problem between the map and pixels (how to enter the pixels to the map) for making frequency table
- After Huffman tree implementation , we do not know that there is a difference between serialization in text file and binary file
- Padding problem ,and if the last byte is incomplete , it will be full with zeros
- At serialization , we do not know if we use array of bytes or up to ofstream in writing

## 5. User manual for the system

./compress "filepath"

./decompress "compressed file path"

## 6. Results

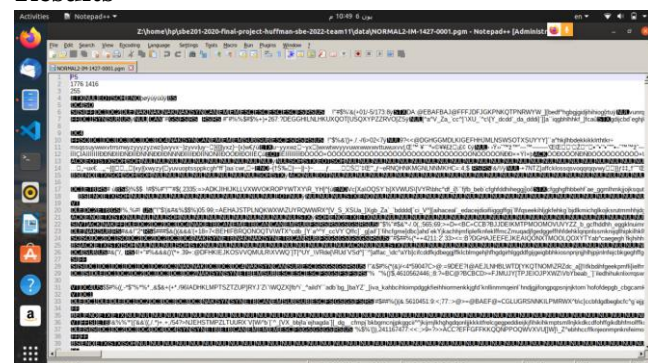
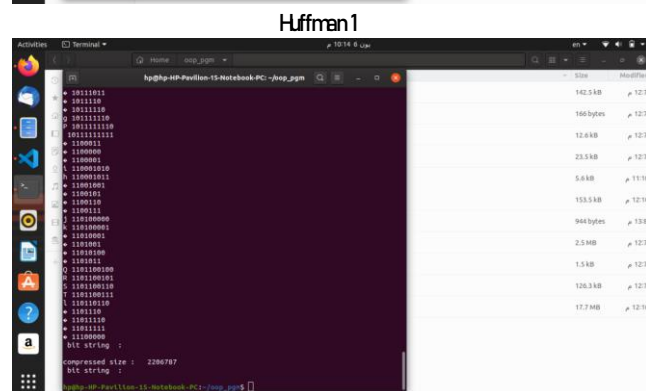
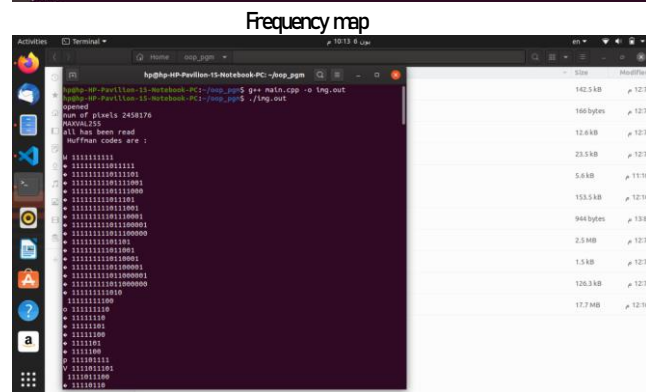
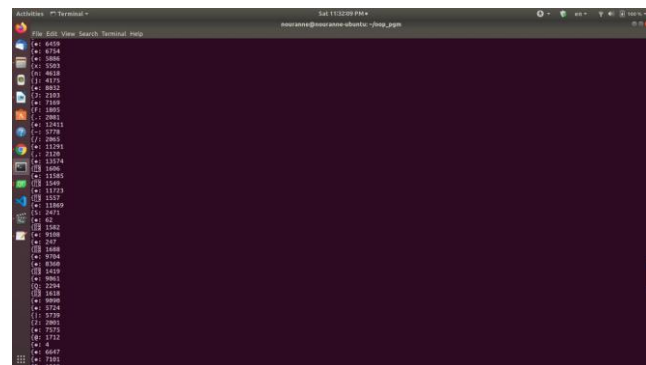
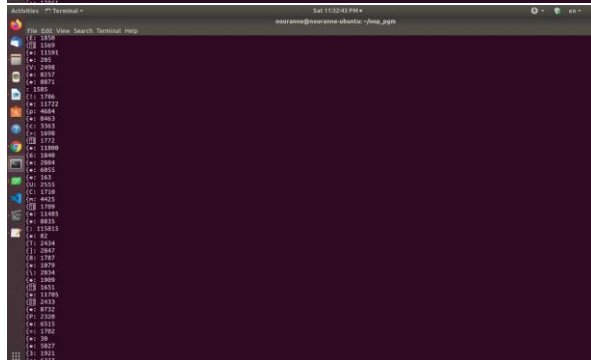
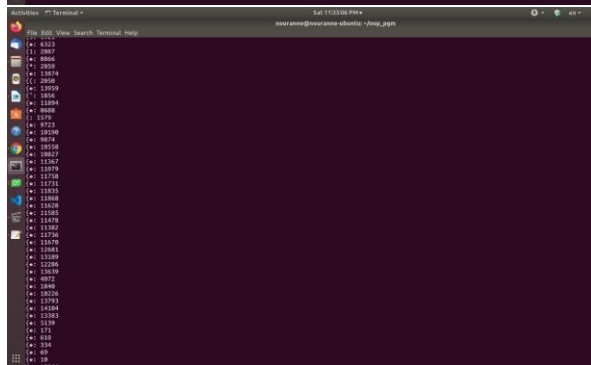
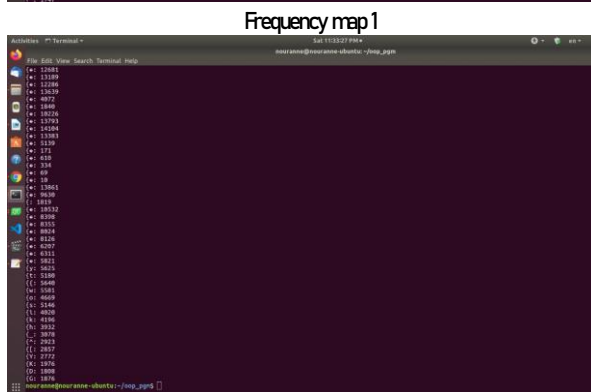
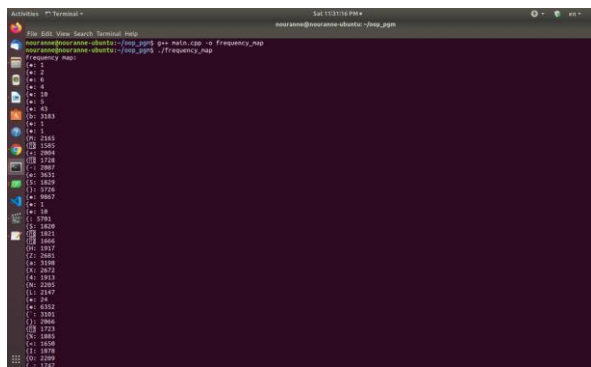
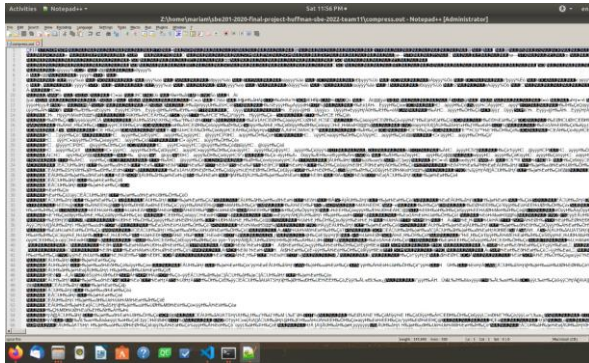


Image reading





serilization

## 7. Contributions

- Reading : nouran khaled & alaa essam
- Frequency map : nouran & Mariam & meeran
- Huffman tree and encoding :maraiam ashraf & meeran ahmed
- Decoding : mariam & aya & meeran
- Serialization : nouran
- Deserialization : alaa
- Report : aya

## 8. References

- [1]"vector - C++ Reference", Cplusplus.com, 2020. [Online]. Available: <http://www.cplusplus.com/reference/vector/vector/>.
- [2] S. Singh and V. kumar Gupta, 2016. [Online]. Available: [https://www.academia.edu/27803932/JPEG\\_Image\\_Compression\\_and-Decompression\\_by\\_Huffman\\_Coding](https://www.academia.edu/27803932/JPEG_Image_Compression_and-Decompression_by_Huffman_Coding).
- [3] "Reference - C++ Reference", Cplusplus.com . [Online]. Available: <http://www.cplusplus.com/reference/> . [Accessed: 06- Jun- 2020].
- [4] "vector - C++ Reference", Cplusplus.com, 2020. [Online]. Available: <http://www.cplusplus.com/reference/vector/vector/>.
- [5]"Free Algorithms Book", Books.goalkicker.com. [Online]. Available: