

Text Compression Using Huffman Compression Algorithm

By Group 6

Apoorva Pendse	12211435
Parth Rajopadhye	12211446
Shreyas Ruikar	12211663
Ifra Shaikh	12210467
Rajeev Tapadia	12210767

INTRODUCTION

- Huffman Coding is a technique of compressing data to reduce its size without losing any of the details.
- Instead of using standard 8-bit size, It adapts bit lengths according to the frequencies in which they occur.
- Shorter codes assigned to frequent characters for efficiency.
- Optimal code generation reduces the required storage space to a minimum.
- Enhanced data compression efficiency.

LOSSY VS LOSSLESS: BENEFITS AND DISADVANTAGES

	LOSSY	LOSSLESS
Pros	<ul style="list-style-type: none">• Small file size• Ideal for serving images on the web• Many tools available for this type of compression• Altered quality usually invisible for the naked eye	<ul style="list-style-type: none">• Zero loss in quality• Slight decrease in file size• Reversible image reconstruction
Cons	<ul style="list-style-type: none">• Quality can be degraded as the rate of compression is higher	<ul style="list-style-type: none">• Compressed files are still large• Can impact the loading speed

OBJECTIVES

GUI DEVELOPMENT

Creating a GUI for Huffman Compression Algorithm.

CLI CREATION

Implementing Huffman Algorithm to Compress and Decompress text.

EFFICIENCY

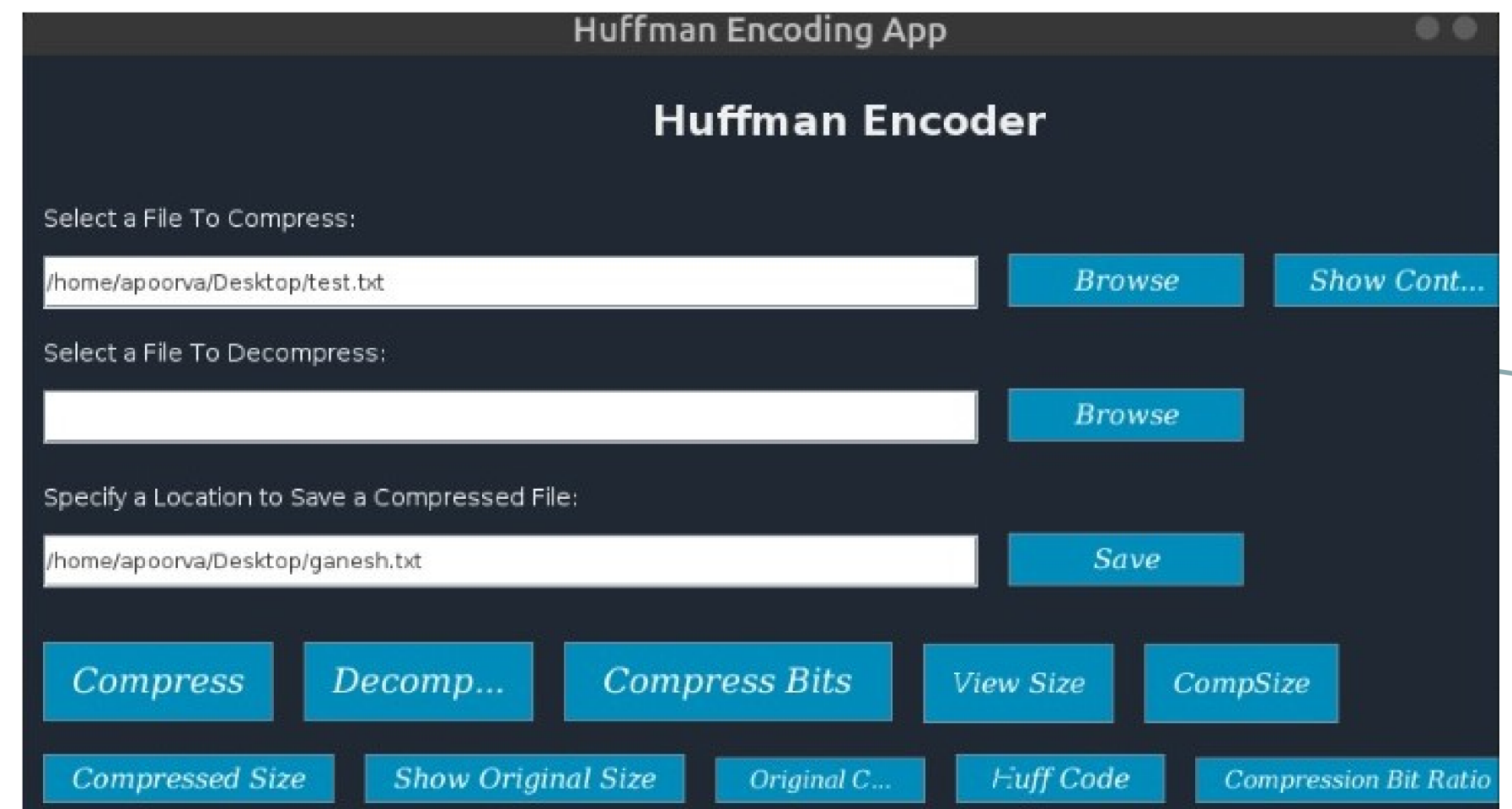
Enabling efficient storage of data without loss.



PROBLEM STATEMENT

This project aims to create a user-friendly GUI for Text File Compression using Huffman Algorithm. It allows users to compress text files while maintaining data integrity, and efficiently store binary sequences of alphabetic characters, enhancing data management.

GUI



The screenshot displays the 'Huffman Encoding App' window. The title bar reads 'Huffman Encoding App'. The main title is 'Huffman Encoder'. The interface includes three input sections: 'Select a File To Compress:' with a text field containing '/home/apoorva/Desktop/test.txt', a 'Browse' button, and a 'Show Cont...' button; 'Select a File To Decompress:' with an empty text field and a 'Browse' button; and 'Specify a Location to Save a Compressed File:' with a text field containing '/home/apoorva/Desktop/ganesh.txt' and a 'Save' button. At the bottom, there are two rows of buttons. The first row contains 'Compress', 'Decomp...', 'Compress Bits', 'View Size', and 'CompSize'. The second row contains 'Compressed Size', 'Show Original Size', 'Original C...', 'Huff Code', and 'Compression Bit Ratio'.

METHODOLOGY

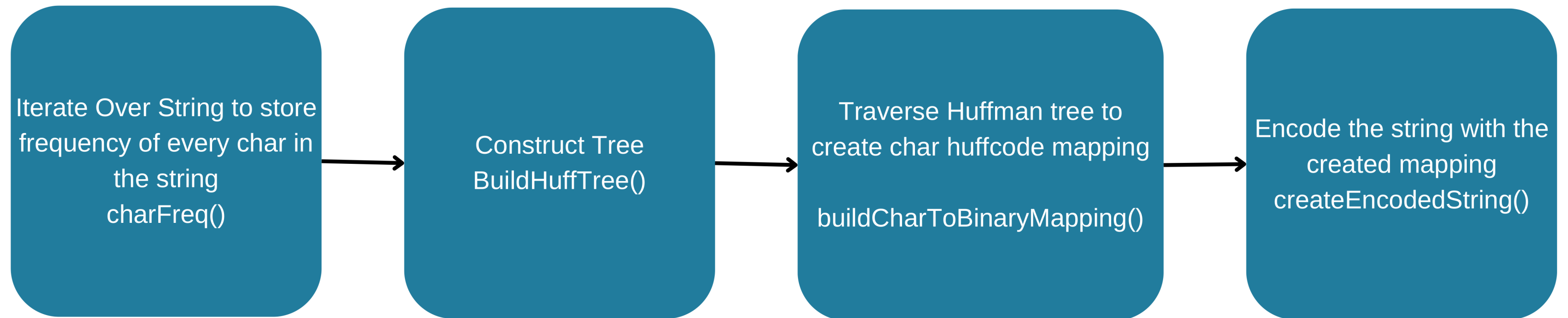
- Gather requirements and create a design plan.
- Write code for the GUI.
- Write code for compression and decompression algorithm.
- Implement the algorithm using GUI.

METHODOLOGY

- Huffman Tree
- Min Heap
- Unordered Map
- Vectors
- Priority Queue

```
class node{  
public:  
    char ch;  
    int freq;  
    node *left;  
    node *right;  
};
```

FLOW DIAGRAM



CONCLUSION

- **Successful creation of a user-friendly Huffman Coding GUI application.**
- **Efficient data compression and decompression while preserving data integrity.**
- **Systematic project methodology ensured a reliable application.**
- **Saved disk space due to reduced file size.**

REFERENCES

- Stanford University (Paper): Huffman Encoding and Data Compression
- A Study on Data Compression Using Huffman Coding Algorithms
–D.Jasmine Shoba, Dr.S Sivakumar
- GeeksForGeeks – Huffman Coding
- Huffman Codes: An Information Theory Perspective
–Reducible



The background features several decorative geometric elements. In the top-left corner, there are thin, parallel diagonal lines. In the top-right corner, there is a cluster of overlapping semi-circles in teal, orange, and red. In the bottom-left corner, there is another cluster of overlapping semi-circles in teal, orange, and red. In the bottom-right corner, there is a large, faint, light-blue circular arc and some thin diagonal lines.

THANK YOU