



PROJECT REPORT

FILE COMPRESSION TOOL



REGISTRATION NUMBER

2023-CS-524

2023-CS-539

2023-CS-553

DECEMBER 16, 2024
DATA STRUCTURE AND ALGORITHM
Dr.Muhammad Idrees

Project Title: File Compression Tool

1. Abstract

This project focuses on developing a file compression tool that implements Huffman encoding to reduce the size of text files. The tool accepts a .txt file, compresses its contents using Huffman compression, and provides users with a downloadable compressed file. The aim of this project is to optimize file storage and transmission by reducing the file size while maintaining data integrity. The web-based tool uses Express.js, Multer for file handling, and custom Huffman compression algorithms.

2. Introduction

File compression plays a crucial role in reducing storage space and optimizing transmission over networks. In this project, the objective was to design a tool that allows users to upload text files, compress them using Huffman coding, and download the compressed version. The compression process ensures minimal data loss and high efficiency by using variable-length codes for characters based on their frequency of occurrence in the text. This report explains the problem, objective, scope, and the working mechanism of the tool.

3. Problem Statement

With the rapid growth of digital content, especially text-based data, file size management has become a major challenge. Large files require more storage space and take longer to transfer over the internet. The problem addressed in this project is to develop a tool that can efficiently compress large text files to a smaller size without significant data loss. The goal is to make the files more manageable for storage and faster for transmission.

4. Objectives

The main objectives of this project are:

- To create a user-friendly web-based tool for file compression.
- To implement the Huffman coding algorithm to compress text files.
- To allow users to upload .txt files, compress them, and download the compressed version.
- To show the original size, compressed size, and compression percentage to the users.
- To provide a responsive, simple, and efficient UI for easy interaction.

5. Scope of the Project

This project focuses on the compression of text files using the Huffman algorithm. The tool only supports .txt files for now and allows users to upload files, compress them, and download the compressed version. The scope of this tool can be expanded in future versions to support other file formats, enhance compression techniques, and integrate additional features such as password protection or encryption. Additionally, the project aims to provide accurate feedback to the user on the effectiveness of the compression process.

6. Significance of the Study

This project is significant in several ways:

- **Storage Optimization:** By compressing files, users can save storage space on their devices or servers.
- **Data Transmission:** Smaller file sizes can be transferred faster over the internet, making it suitable for email attachments, cloud storage, and file sharing services.
- **Efficiency:** The tool ensures efficient use of storage and faster data handling, particularly for large text files.
- **Educational Value:** This project demonstrates the implementation of a classic data compression algorithm, which is a key concept in computer science and data processing.

7. Methodology

The project is built using **Node.js** with **Express.js** for backend services. The file upload is handled by **Multer**, and the Huffman compression algorithm is implemented in JavaScript. The following steps describe the process:

1. **File Upload:** Users upload .txt files through a form on the web interface.

Huffman Compression:

- **Frequency Map:** A frequency map is built based on the occurrence of each character in the text file.
- **Huffman Tree:** A binary tree is constructed using the frequency map, where each leaf node represents a character and its frequency.
- **Huffman Codes:** Each character is assigned a binary code based on the structure of the Huffman tree.
- **Data Encoding:** The input text is encoded into a binary string using the generated Huffman codes.

2. **Compression:** The binary string is converted into a buffer that represents the compressed file.
3. **User Feedback:** The original file size, compressed file size, and compression percentage are calculated and displayed to the user.

8. Results

Upon successfully uploading a .txt file, the tool provides the following results:

- **Original Size:** Displays the size of the uploaded file before compression.
- **Compressed Size:** Displays the size of the file after Huffman compression.
- **Compression Percentage:** Shows the percentage reduction in file size after compression.

The compressed file is made available for download in .huff format, allowing users to retrieve the smaller version of their file.

For example:

- **Original Size:** 1500 bytes
- **Compressed Size:** 800 bytes
- **Compression Percentage:** 46.67%

9. Conclusion

The **File Compression Tool** successfully implements the Huffman encoding algorithm to compress .txt files. It provides a simple web interface for uploading and downloading compressed files, along with key information about file size reduction. The project demonstrates the practical application of data compression algorithms and their impact on storage and data transfer efficiency.

Future improvements could include adding support for other file formats, enhancing compression efficiency, and integrating more advanced techniques like lossless or lossy compression. Additionally, the user interface could be further improved to provide a more interactive experience, including real-time feedback on compression progress.

10. References

- "Huffman Coding" – Wikipedia. https://en.wikipedia.org/wiki/Huffman_coding
 - "Node.js Documentation" – <https://nodejs.org/en/docs/>
 - "Express.js Documentation" – <https://expressjs.com/>
 - "Multer Documentation" – <https://www.npmjs.com/package/multer>
-

