## Final Report On

# Compresspro

**Prepared by**

Md. NASIM MOLLA
MUH2025024M
Year-03, Term-01
***E-mail:*** *nasim2515@student.nstu.edu.bd*

**Supervised by**

Md. Eusha Kadir
Lecturer
Institute of Information Technology
Noakhali Science and Technology University

**Submission Date: 09.07.23**

07-09-2023

# Contents

# Compresspro

## 1.Abstract

The "Compresspro" project is a Java-based compression and decompression tool that allows users to choose between two algorithms: Huffman Coding and LZW Compression. This report outlines the project's design, implementation, usage instructions, testing, and results.

## 2. Introduction

Data compression is essential in various fields to reduce storage requirements and improve data transmission efficiency. The "Compresspro" project aims to provide an easy-to-use compression and decompression tool with support for two popular compression algorithms: Huffman Coding and LZW Compression.

## 3.Objectives

1) The main objectives of the project are as follows: Implement the Huffman Coding algorithm for data compression and decompression.
2) Implement the LZW Compression algorithm for data compression and decompression.
3) Develop a user-friendly command-line interface for compression and decompression operations.
4) Evaluate the performance and effectiveness of both compression algorithms.

## 4. Project Scope

The scope of the project includes the implementation of two compression and decompression algorithms and the development of a command-line interface for users to interact with the tool. The project does not cover graphical user interfaces or additional features beyond compression and decompression

## 5. Project Design

### Architecture Overview

The project follows a modular design, separating the core compression and decompression algorithms from the user interface. The architecture consists of the following components:

There are five class in my project. They are as follows: -

**Main.java:** This module interacts with users. User can use this class for file compressing and decompressing.

**HuffmanNode.java:** This class define node needed for Huffman coding algorithm.

**HuffmanTree.java:** This class construct tree for compression and also contain method for decoding.

**LzwEncoding**: This class compress the given file and return a encoded code.

**LzwDecoding**: This class decompress the code into original file and return the original file.

## 6. Implementation

### Language and Tools: -

Programming Language: Java with OOP
Development Environment: NetBeans
Data Structures

- Hash Map: Stores key-value pairs for substring substitution.
- Array List: Stores encoded and decoded data dynamically.
- StringBuilder: Uses for mutable string array.
- Priority Queue: Stores data based on priority.

## 7. Usage Instructions

Explain how users can compress data using both Huffman and LZW algorithms.
Explain how users can decompress compressed data using both Huffman and LZW algorithms.

## 8. Testing

### Test Cases

Describe the test cases used to evaluate the correctness of the compression and decompression processes.

Performance Evaluation

**Algorithm Type:**

Huffman Coding: Huffman is a variable-length prefix coding algorithm. It assigns shorter codes to more frequent symbols in the input data.

LZW Algorithm: LZW is a dictionary-based compression algorithm that replaces repeating sequences of characters with shorter codes.
Compression Efficiency:

**Compression Efficiency:**

Huffman Coding: Huffman is efficient for compressing data with varying symbol frequencies. It works well when some symbols are more frequent than others.

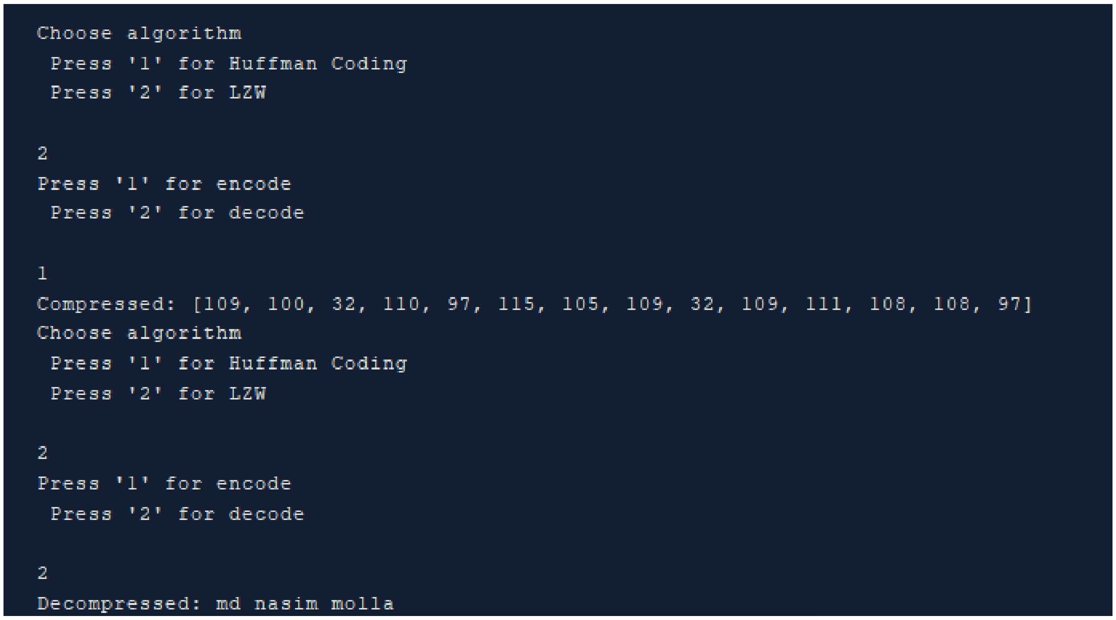
LZW Algorithm: LZW is effective for compressing data with repeating patterns, such as text data or certain types of binary data.
Dictionary Usage:

**Adaptability:**

Huffman Coding: Huffman is not adaptive; the tree structure is fixed based on input statistics.

LZW Algorithm: LZW is adaptive; it dynamically updates its dictionary as it encounters new patterns in the input data.

## 9. Results/Work flow



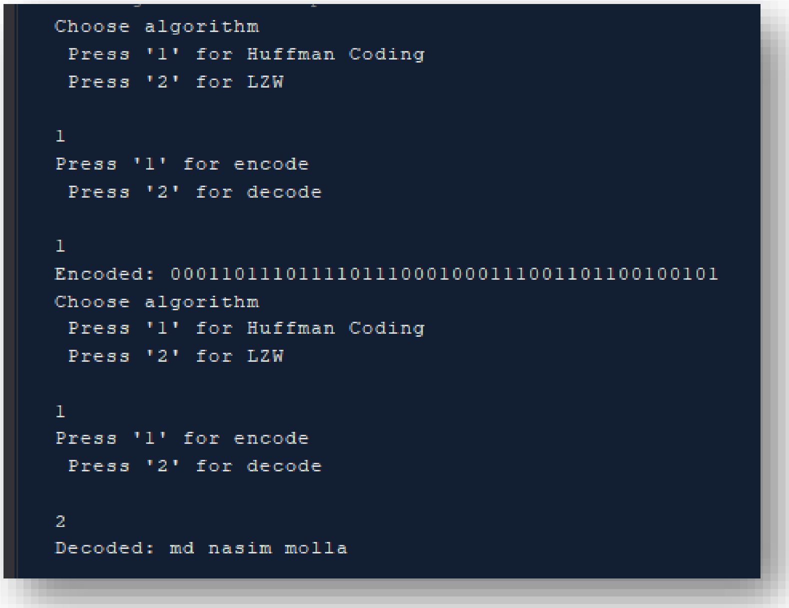
```
Choose algorithm
 Press '1' for Huffman Coding
 Press '2' for LZW

2
Press '1' for encode
 Press '2' for decode

1
Compressed: [109, 100, 32, 110, 97, 115, 105, 109, 32, 109, 111, 108, 108, 97]
Choose algorithm
 Press '1' for Huffman Coding
 Press '2' for LZW

2
Press '1' for encode
 Press '2' for decode

2
Decompressed: md nasim molla
```



```
Choose algorithm
 Press '1' for Huffman Coding
 Press '2' for LZW

1
Press '1' for encode
 Press '2' for decode

1
Encoded: 00011011101111101110001000111001101100100101
Choose algorithm
 Press '1' for Huffman Coding
 Press '2' for LZW

1
Press '1' for encode
 Press '2' for decode

2
Decoded: md nasim molla
```

## 10. Future Improvements

I will develop a user-friendly interface for "Compresspro" tool. Further I will learn another algorithm like, Lz77 and Gzip algorithm and include into my software tool.

## 11. Conclusion

I have successfully learned and implement Huffman coding and Lzw compression and decompression algorithm. I faced challenges during working with this project such as time management and learn those algorithms.