HUFFMAN CODING

TEXT COMPRESSION

IMAGE COMPRESSION

62	TARUNKUMAR S
71	RITHISH BARATH N
84	RITHISH K

INTRODUCTION

1️⃣ Huffman Coding is a lossless data compression algorithm that assigns shorter codes to frequently occurring characters, reducing file size efficiently. It is based on the concept of variable-length prefix codes.

2️⃣ It plays a crucial role in optimizing storage and bandwidth while maintaining data integrity in compressed files. By minimizing redundancy, it ensures efficient data encoding.

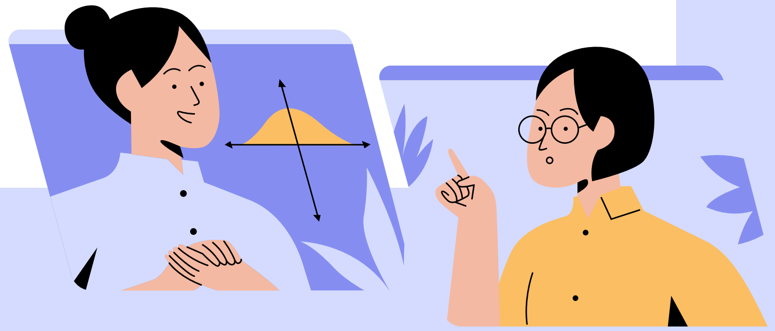
3️⃣ Widely used in applications like ZIP, GZIP, JPEG, MP3, and MPEG, it enhances data transmission and storage efficiency. Its adaptability makes it a fundamental technique in modern compression algorithms.

OBJECTIVES

Here are three main objectives of Huffman Coding:
1️⃣ Efficient Data Compression – Reduces the size of text and image files by assigning shorter codes to frequently used symbols, minimizing storage requirements.

2️⃣ Lossless Compression – Ensures that no data is lost during compression and decompression, maintaining the original information perfectly.

3️⃣ Optimized Data Transmission – Speeds up data transfer by reducing the number of bits required, improving efficiency in networks and storage systems.



RULES

1️⃣ Characters with higher frequency are assigned shorter binary codes.

2️⃣ A binary tree (Huffman Tree) is constructed based on character frequencies.

3️⃣ Each character’s code follows the prefix property (no code is a prefix of another).

4️⃣ The tree is built using a greedy algorithm by merging the least frequent nodes first. Encoding and decoding follow the unique binary paths from the root of the tree.

TIME AND SPACE COMPLEXITY

1️⃣ Best Case (Ω)

If all characters have the same frequency, the Huffman tree is balanced.

Time Complexity: $\Omega(n \log n)$

Space Complexity: $O(n)$

2️⃣ Average Case (Θ)

In most practical scenarios, frequencies vary, and the tree structure depends on input distribution.

Time Complexity: $\Theta(n \log n)$

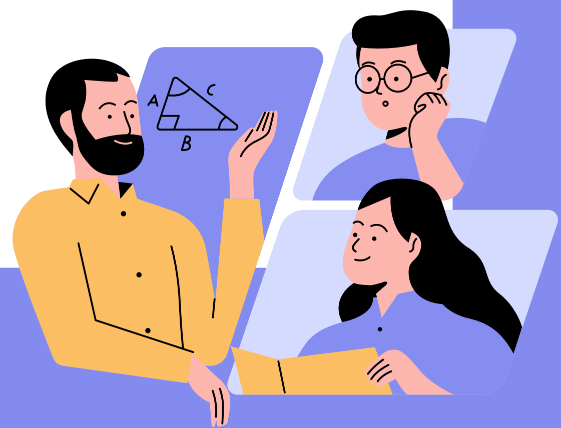
Space Complexity: $O(n)$

3️⃣ Worst Case (O)

If one character is extremely frequent and others are rare, the tree becomes skewed.

Time Complexity: $O(n \log n)$

Space Complexity: $O(n)$



REAL TIME APPLICATIONS

1️⃣ File Compression (ZIP, GZIP, BZIP2)

2️⃣ Image Compression (JPEG , PNG)

3️⃣ Audio & Video Compression (MP3, MPEG, AAC)

4️⃣ Fax Transmission & Telecommunication

5️⃣ Compiler Design & Data Encoding

FILE COMPRESSION (ZIP, GZIP, BZIP2)

Huffman coding is widely used in lossless file compression algorithms like ZIP, GZIP, and BZIP2. It reduces file sizes by assigning shorter binary codes to frequently occurring characters and longer codes to less frequent ones.

How It Works in File Compression:

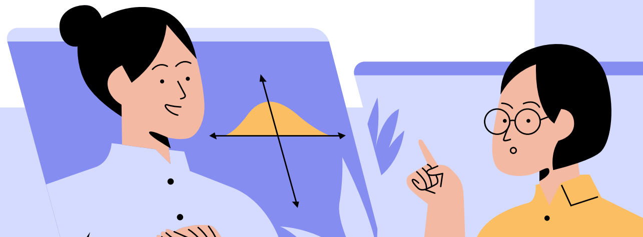
1️⃣ Analyzes the frequency of characters in the file.

2️⃣ Builds a Huffman tree and assigns variable-length codes.

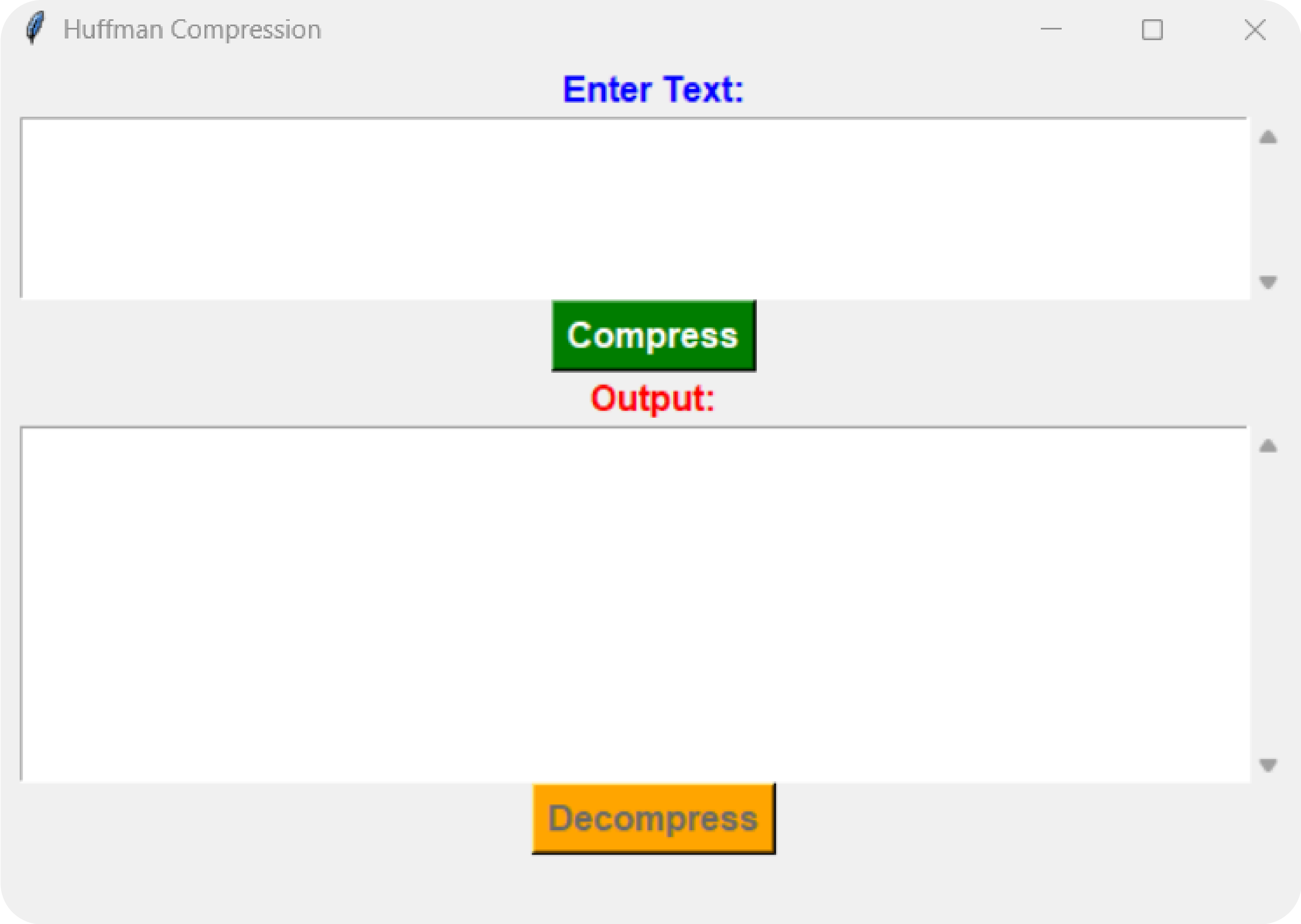
3️⃣ Replaces characters with their respective binary codes.

4️⃣ Compressed file is stored efficiently, reducing storage space and transmission time.

It ensures fast decompression and is widely used in data storage and transmission applications.



FILE COMPRESSION (ZIP, GZIP, BZIP2)



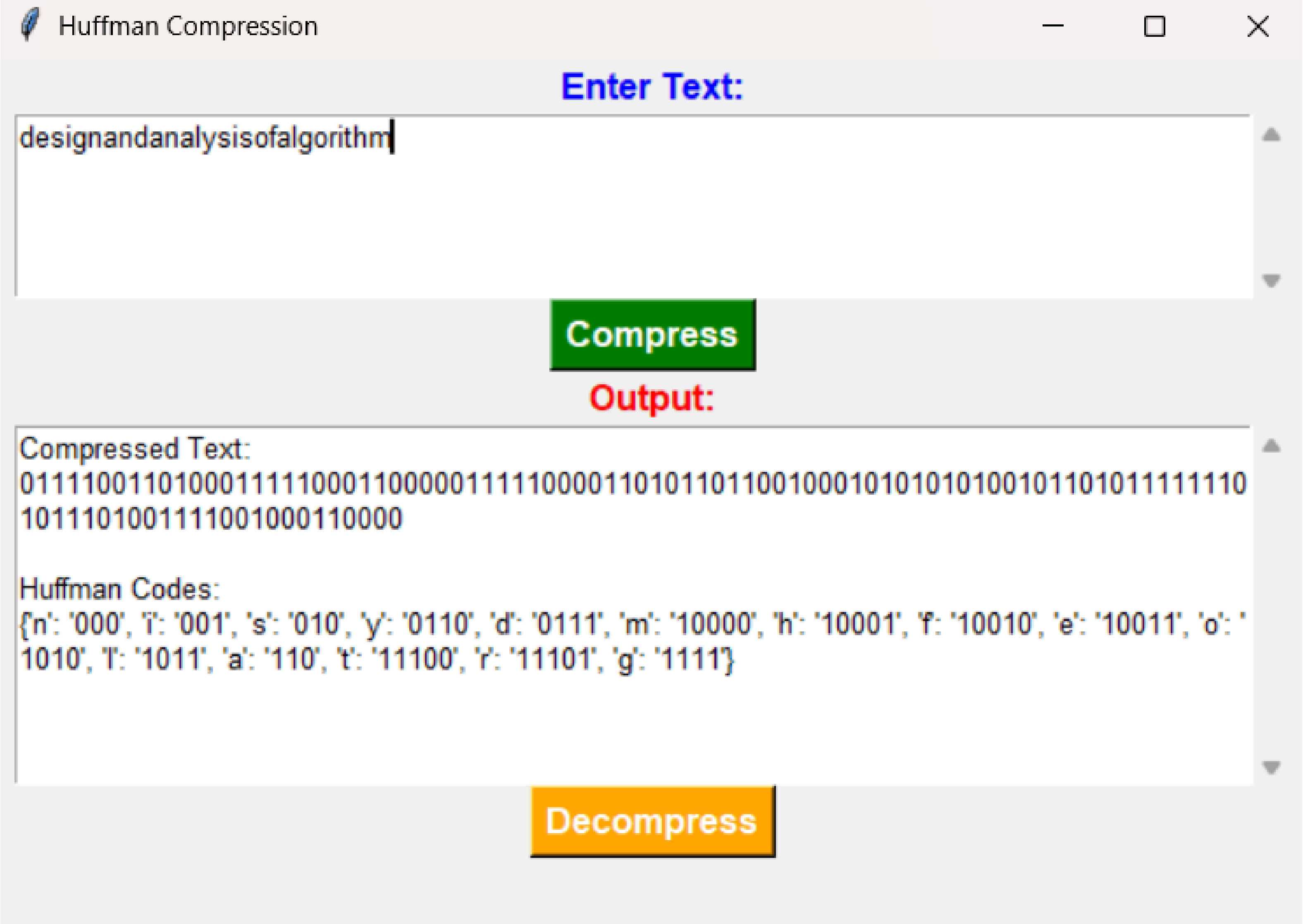
Huffman Compression

Enter Text:

Compress

Output:

Decompress



Huffman Compression

Enter Text:

designandanalysisofalgorithm

Compress

Output:

```
Compressed Text:
01111001101000111110001100000111110000110101101100100010101010100101101011111110
1011101001111001000110000

Huffman Codes:
{'n': '000', 'i': '001', 's': '010', 'y': '0110', 'd': '0111', 'm': '10000', 'h': '10001', 'f': '10010', 'e': '10011', 'o': '
1010', 'l': '1011', 'a': '110', 't': '11100', 'r': '11101', 'g': '1111'}
```

Decompress

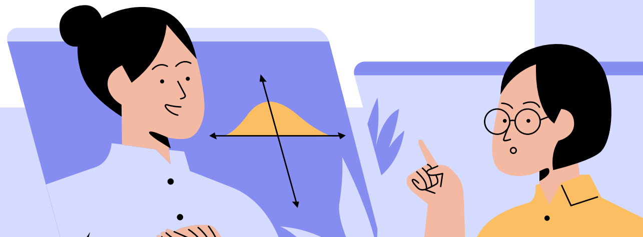


IMAGE COMPRESSION (JPEG , PNG)

Huffman coding is used in lossless image compression algorithms like JPEG and PNG to efficiently reduce file sizes while preserving image quality.

How It Works in Image Compression:

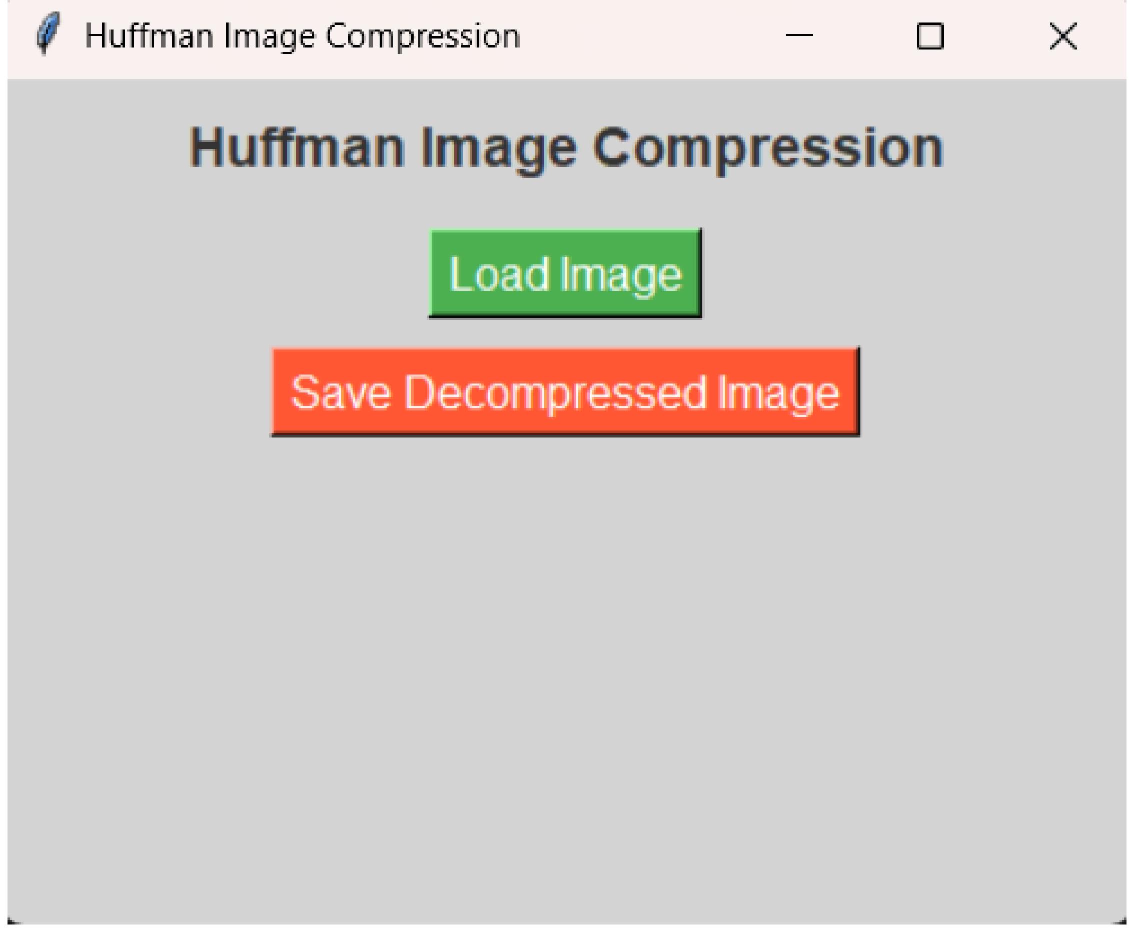
1️⃣ Transforms image data using Discrete Cosine Transform (DCT) or other techniques.

2️⃣ Analyzes frequency of pixel values or coefficient blocks.

3️⃣ Builds a Huffman tree and assigns shorter codes to frequently occurring values.

4️⃣ Encodes the image using these compressed binary codes, reducing storage size.

IMAGE COMPRESSION (JPEG , PNG)



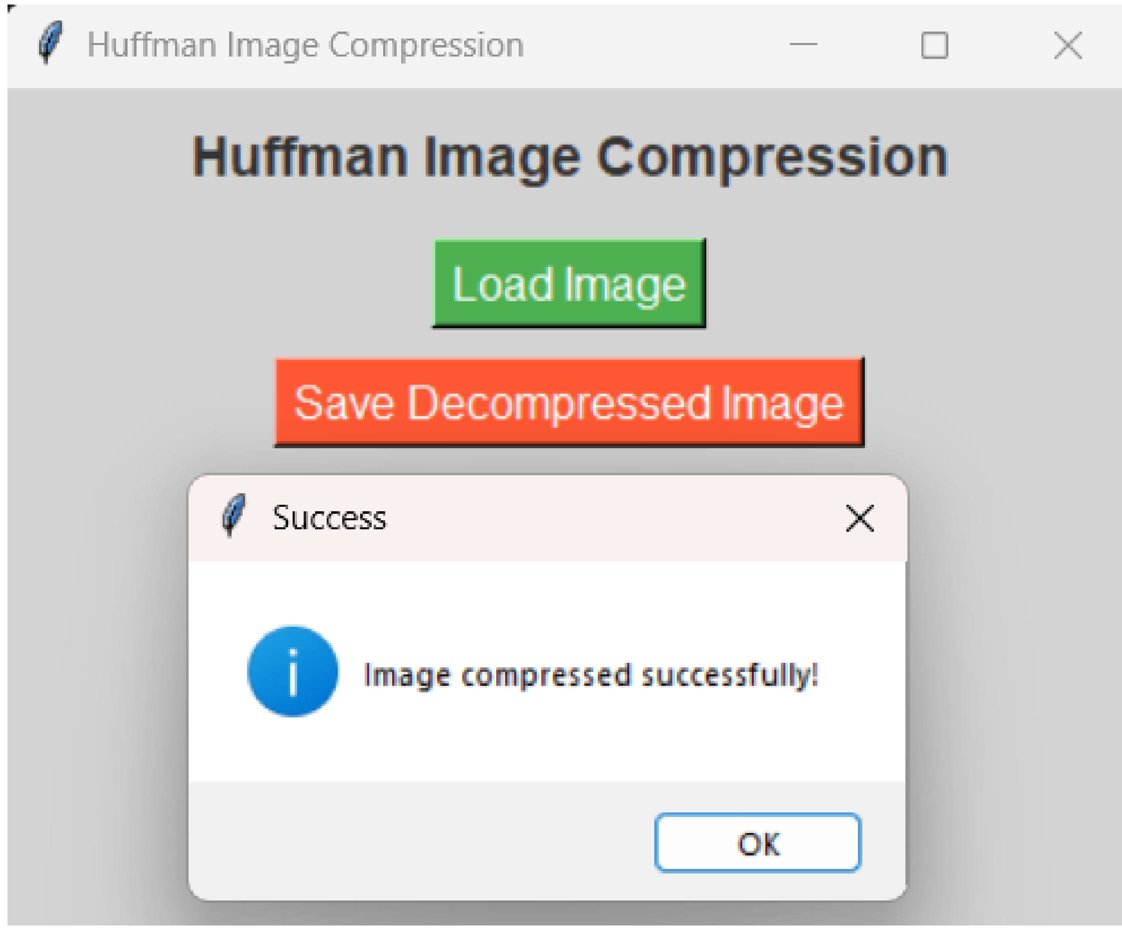


Image compressed successfully!

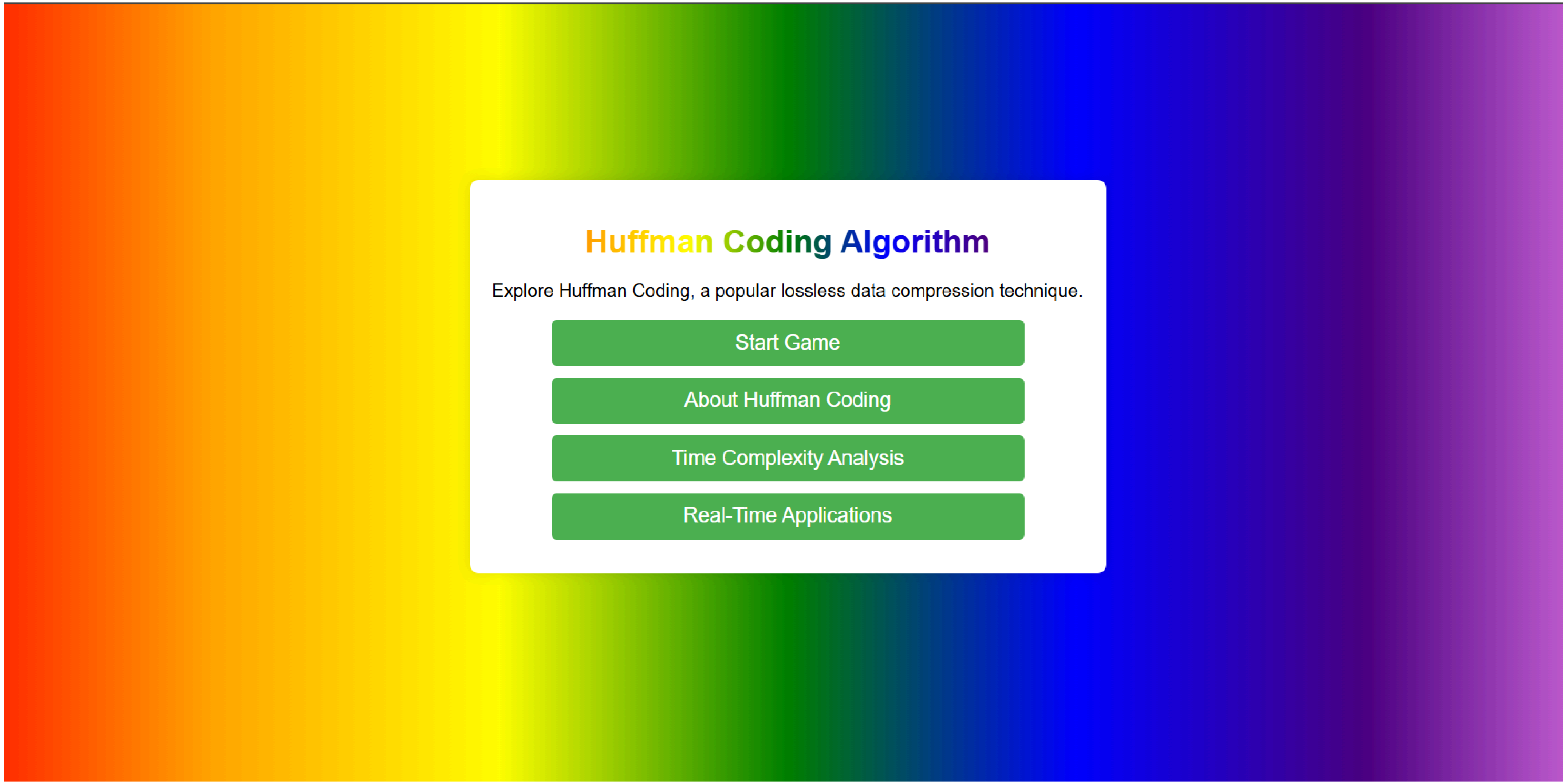
Third Semester students report

Your response has been recorded.

This form was created inside of SRM Institute of Science and Technology.
Does this form look suspicious? Report

Google Forms

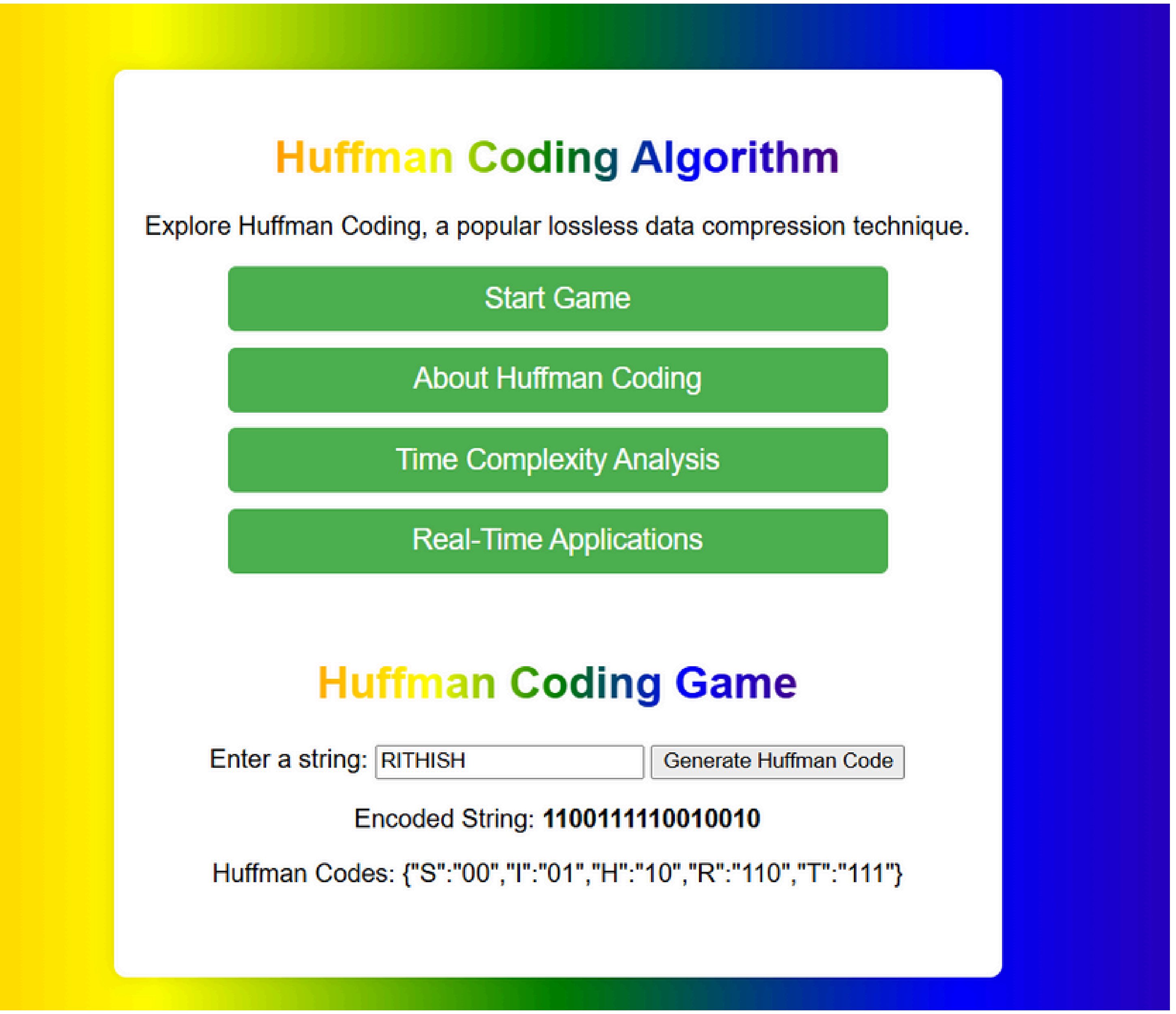
SAMPLE WEBSITE



Huffman Coding Algorithm

Explore Huffman Coding, a popular lossless data compression technique.

SAMPLE WEBSITE



Huffman Coding Algorithm

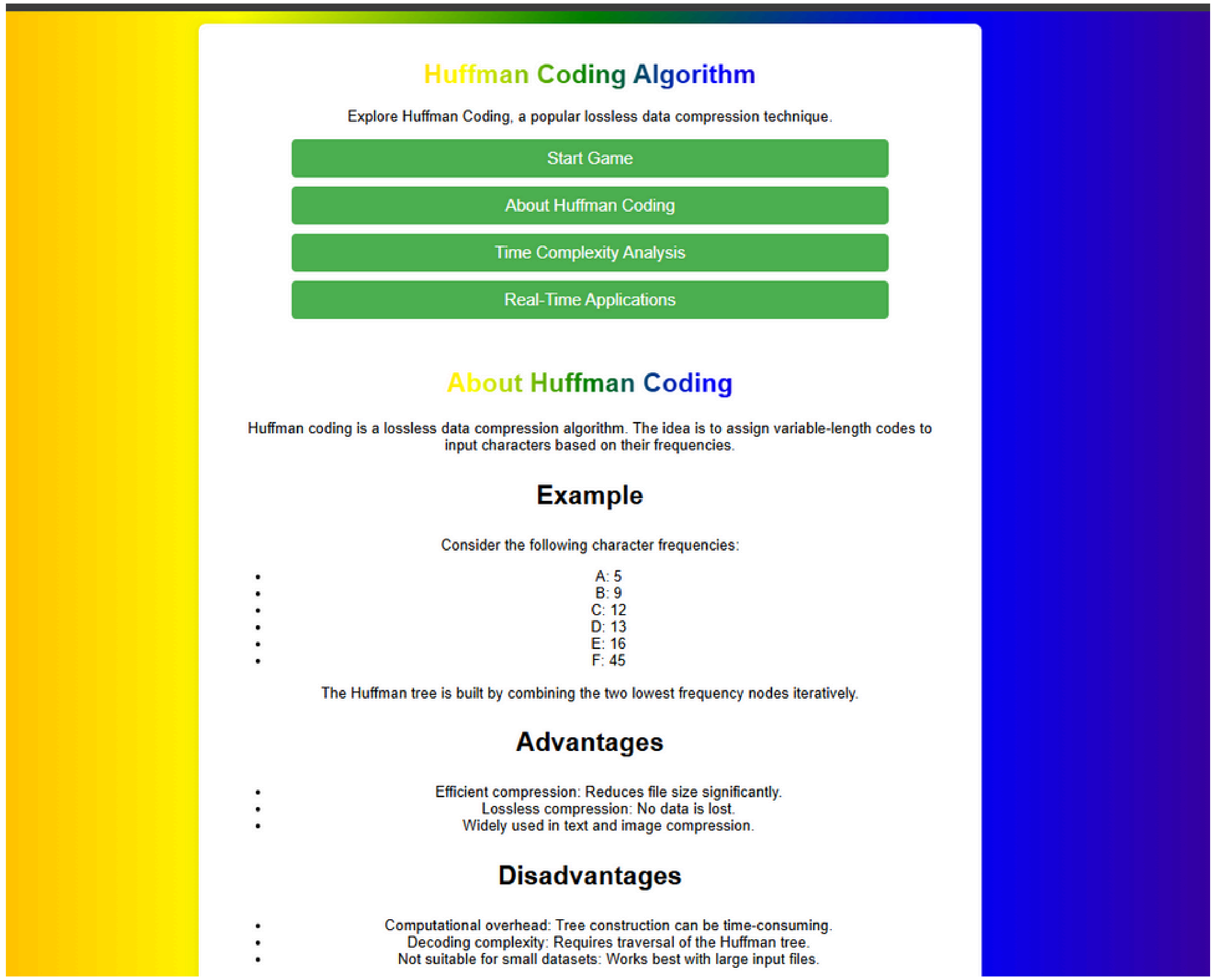
Explore Huffman Coding, a popular lossless data compression technique.

Huffman Coding Game

Enter a string: RITHISH

Encoded String: **1100111110010010**

Huffman Codes: {"S":"00","I":"01","H":"10","R":"110","T":"111"}



Huffman Coding Algorithm

Explore Huffman Coding, a popular lossless data compression technique.

About Huffman Coding

Huffman coding is a lossless data compression algorithm. The idea is to assign variable-length codes to input characters based on their frequencies.

Example

Consider the following character frequencies:

- A: 5
- B: 9
- C: 12
- D: 13
- E: 16
- F: 45

The Huffman tree is built by combining the two lowest frequency nodes iteratively.

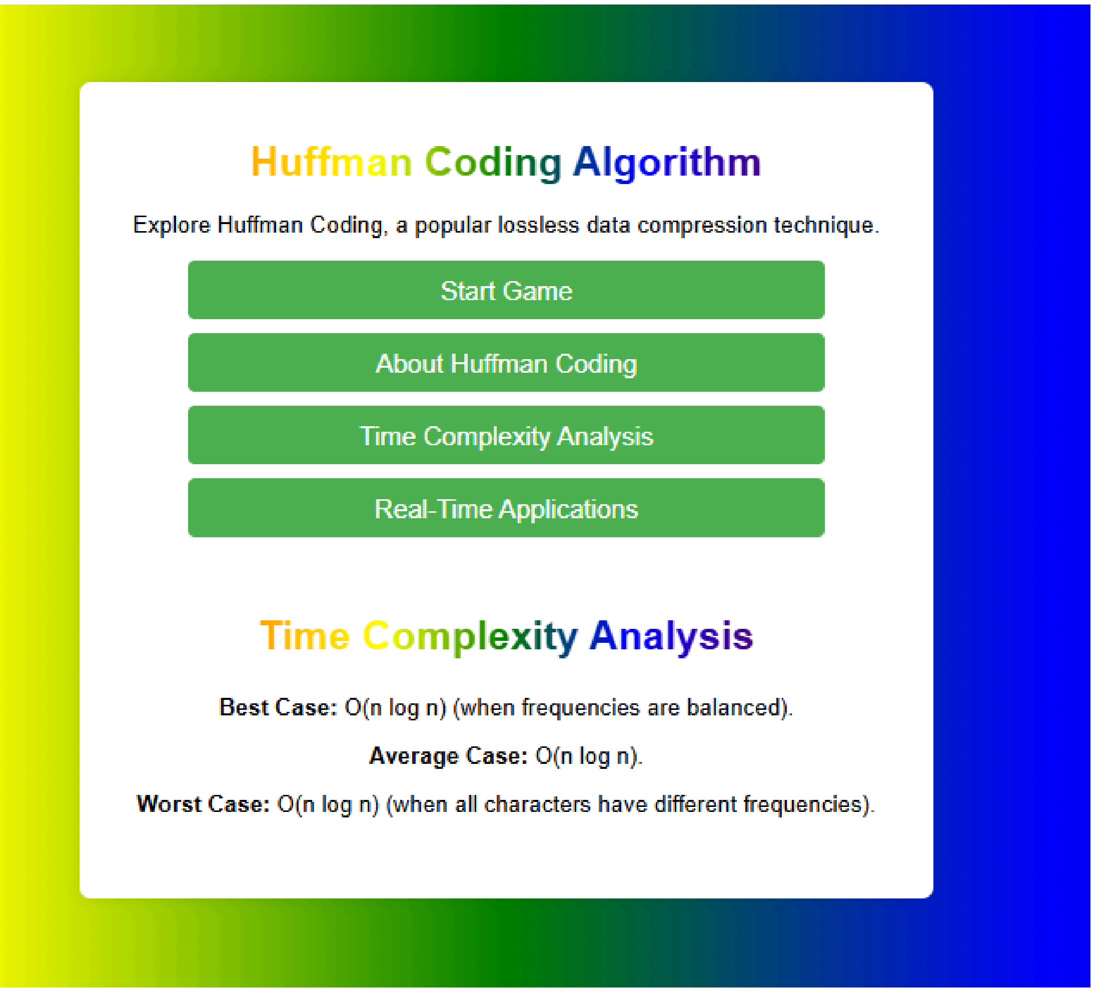
Advantages

- Efficient compression: Reduces file size significantly.
- Lossless compression: No data is lost.
- Widely used in text and image compression.

Disadvantages

- Computational overhead: Tree construction can be time-consuming.
- Decoding complexity: Requires traversal of the Huffman tree.
- Not suitable for small datasets: Works best with large input files.

SAMPLE WEBSITE



Huffman Coding Algorithm

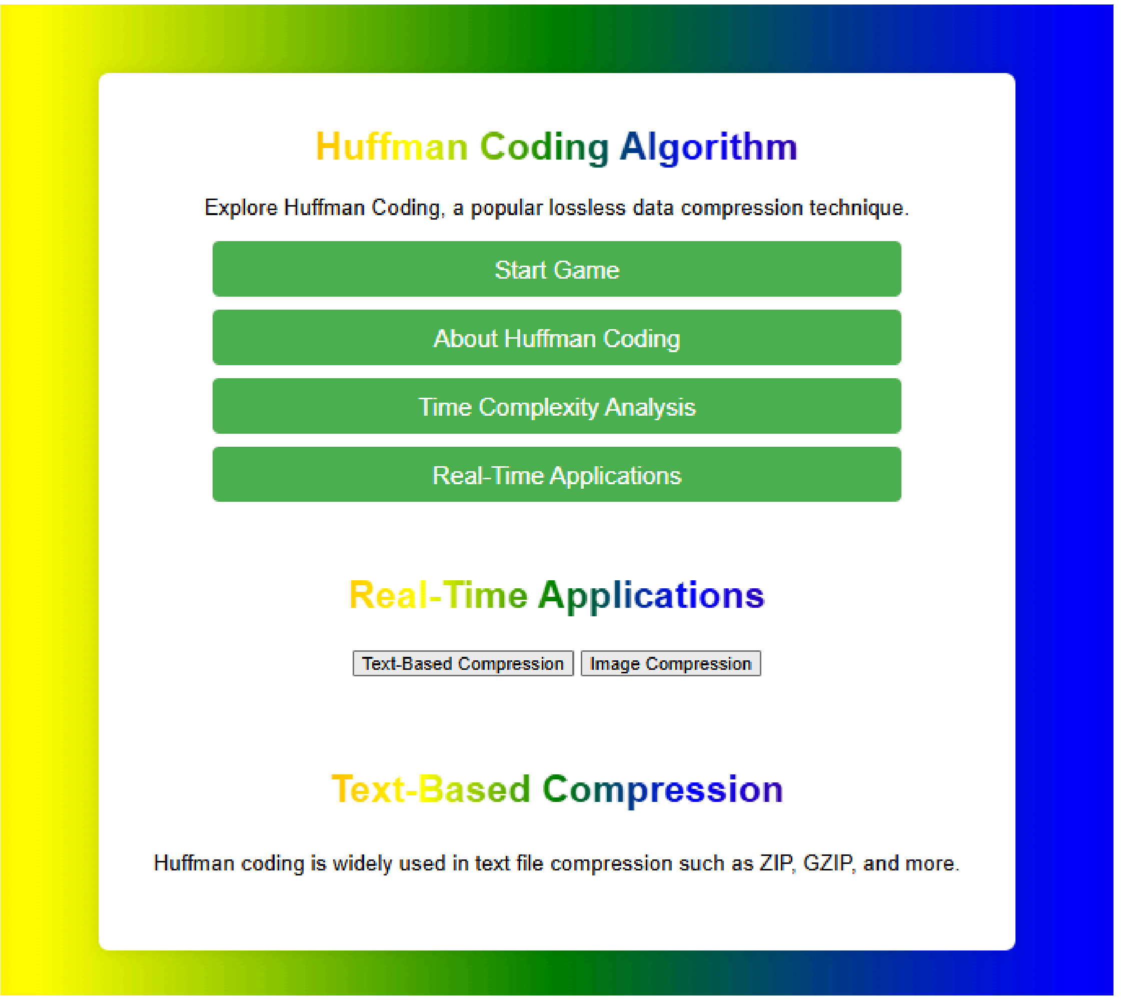
Explore Huffman Coding, a popular lossless data compression technique.

Time Complexity Analysis

Best Case: O(n log n) (when frequencies are balanced).

Average Case: O(n log n).

Worst Case: O(n log n) (when all characters have different frequencies).



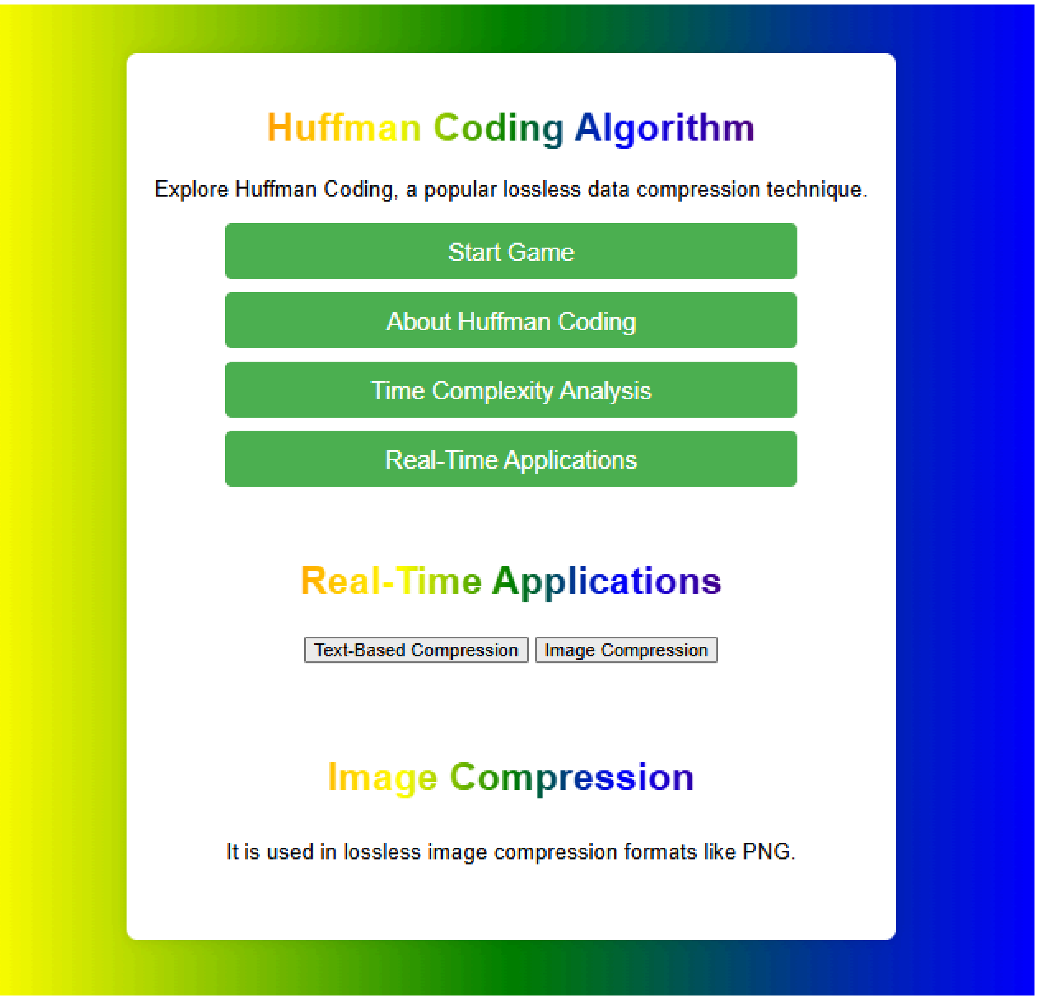
Huffman Coding Algorithm

Explore Huffman Coding, a popular lossless data compression technique.

Real-Time Applications

Text-Based Compression

Huffman coding is widely used in text file compression such as ZIP, GZIP, and more.



Huffman Coding Algorithm

Explore Huffman Coding, a popular lossless data compression technique.

Real-Time Applications

Image Compression

It is used in lossless image compression formats like PNG.

CONCLUSION

1️⃣ Huffman Coding is a widely used lossless data compression algorithm that minimizes storage space by assigning variable-length codes.

2️⃣ It follows a greedy approach to build an optimal binary tree based on character frequency, ensuring efficient encoding.

3️⃣ The prefix property prevents ambiguity in decoding, making it reliable for data compression.

4️⃣ Huffman Coding is extensively used in text, image, audio, and video compression, improving storage and transmission efficiency.

5️⃣ It is implemented in real-world applications like ZIP, JPEG, MP3, and fax transmission for optimizing data handling.

6️⃣ Despite its efficiency, Huffman Coding is best suited for static data and is often combined with other techniques for enhanced compression.