# Exploring Huffman Encoding

Praneeth Chandra Thota
Naveennarayanan Meyyappan

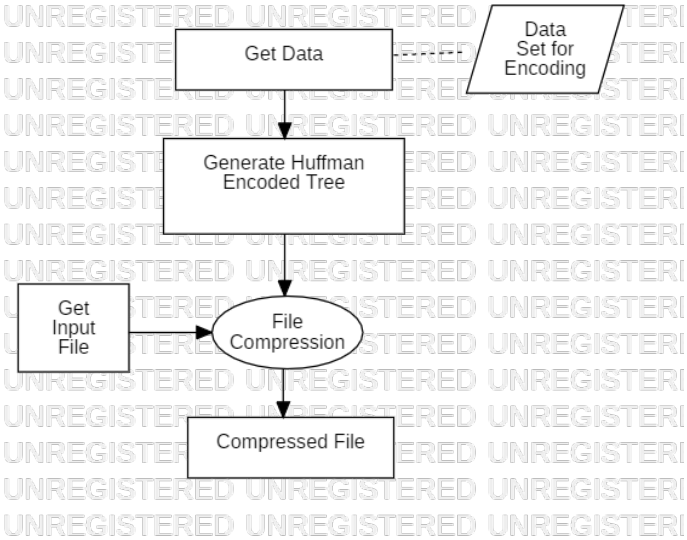
***Abstract*— Most of the applications need storage of large volumes of data and these large volumes of data require a lot of storage. Compressing data reduces the amount of data need to be stored and there by reducing storage costs. This project aims toward the implementation of Huffman Encoding Algorithm to a Data set and reusing the Encoded tree for the compression of other file of same type.**

## I. Project Description

This is an algorithm exploration project in which the Huffman Encoding algorithm is implemented on a data set and the generated encoded tree is reused on other files of same type and a comparison is made on compression ratio from this project with the compression ratio of optimal Huffman method. The usual compression will generate an encoded tree whenever a file is compressed. This project proposes to reuse the encoded tree for file compression so that the time utilized for generating a encoded tree will be saved. And also the length of the header will be reduced. Although this method works faster the compression may not be optimal. We are trying to explore the trade of between the compression ratio and time taken for the compression.

## II. Design

The data is extracted from the data set and the data is run through the Huffman encoding algorithm and the encoded tree is generated. This encoded tree is reused for compressing files of similar type. The following is the flow diagram of the project.



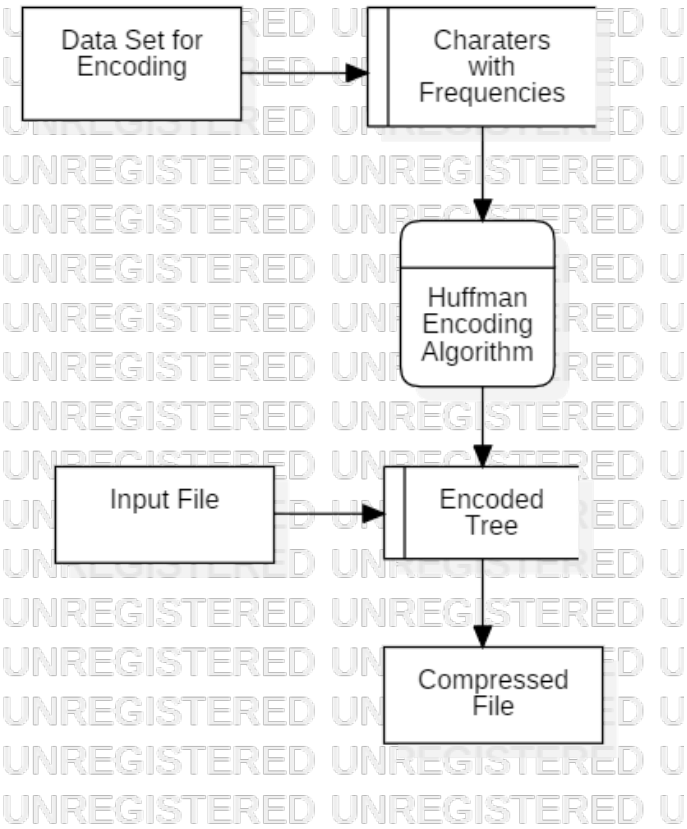

From the data set we get the data and implement Huffman encoding algorithm to generate encoded tree. We use this tree and another input file of the same type which is needed to be compressed as inputs for the file compression algorithm. Finally the compressed file is generated. The transformation of data is shown in the data flow diagram below.

High Level Pseudo Code System Description:

- Generate frequencies of each symbol from the given data set.
- Implement Huffman encoding algorithm on this data.
- Generate an encoded tree.
- Use this tree to compress files with similar type.

Algorithms and Data Structures:

- The main algorithm used in this project is Huffman Encoding Algorithm.
- The data structure used in this project is priority queue.




## III. Challenges

The biggest challenge while implementing this project is coming across the character that is already not present in the encoded tree. We can overcome this challenge by,

- Choosing a data set with wide range of characters involved.
- Whenever a new character is identified the encoded tree can be modified, this method may seem little redundant but in long run it will be efficient. We assume that for each new input at least one new character is identified. As the number of inputs are increased the character range of tree will also be increased.

## IV. References


1) K Ashok Babu, V Satish Kumar, Implementation of data compression using Huffman coding, 2010 International Conference on Methods and Models in Computer Science (ICM2CS-2010).
2) Huffman Encoding and Data Compression, Handout by Julie Zelenski, Keith Schwarz.