



HUFFMAN COMPRESSION

1.DESCRPTION:

Huffman coding is a method of data compression that is independent of the data type, that is, the data could represent an image, audio or spreadsheet. This compression scheme is used in JPEG and MPEG-2. Huffman coding works by looking at the data stream that makes up the file to be compressed. Those data bytes that occur most often are assigned a small code to represent them (certainly smaller than the data bytes being represented). Data bytes that occur the next most often have a slightly larger code to represent them. This continues until all of the unique pieces of data are assigned unique code words. For a given character distribution, by assigning short codes to frequently occurring characters and longer codes to infrequently occurring characters, Huffman's minimum redundancy encoding minimizes the average number of bytes required to represent the characters in a text. Static Huffman encoding uses a fixed set of codes, based on a representative sample of data, for processing texts. Although encoding is achieved in a single pass, the data on which the compression is based may bear little resemblance to the actual text being compressed.

2.DATA STRUCTURES:

The data structure of Huffman codes and its application to efficient encoding and decoding of Huffman codes are studied in detail:

- Min heap (tree structure):
- by class for each node to collect the node char and its code, then we use a data structure “priority queue” as a sorting technique to get min two nodes.
- HashMap: using two hash maps to store codes of Huffman and its frequencies during compression and decompression.

3.ALGORITHM:

The used algorithm is Greedy Algorithm and its complexity Is:

The time complexity of the Huffman algorithm is **$O(n \log n)$** , for the heap to store the weight of each tree, each iteration requires **$O(\log n)$** time to determine the cheapest weight and insert the new weight. There are **$O(n)$** iterations, one for each item.

4.HEADER FORMAT:

some initial information needs to be stored in the compressed file that will be used by the decompression. The tree used for compression is needed to decompress but we can store This information in alternative ways through our header format:

- **Format :**
- The unique characters are stored with their corresponding Huffman code and frequency also the number of bits used for padding