Huffman Encoding/Decoding

Mihir Gajjar (1401076) Vidit Shah (1401078)

Brief Description: Huffman algorithm is a lossless data compression algorithm.

It is Prefix Code. Huffman coding assigns variable length code words to fixed length input characters based on their frequencies. More frequent characters are assigned shorter code words and less frequent characters are assigned longer code words. A Huffman tree is made for the input string and characters are decoded based on their position in the tree. We assign codes to the leaf nodes which represent the input characters.

List of Data Structures Used:

1. Minimum Priority Queue:

Importance: It is used to sort the nodes of the characters of nonzero frequency.

Operation:

**Insertion**: When we insert the nodes in the Minimum Priority Queue then it *sorts the Nodes* in such a way that every parent node’s frequency is less than its child’s node’s frequency. Also the root has the node with minimum frequency.

**Deletion**: According to the property of the minimum priority Queue when we perform the deletion operation then we get the *node with minimum frequency.*

So we perform two deletion operations 🡪 add the frequency of the two (deleted) nodes and make a new node 🡪 Insert that node in the Minimum Priority Queue.

1. Array :

Importance: Using array of nodes all the *nodes can be easily accessed.* Using Array of Nodes we are creating nodes of only those characters who have non zero frequency.

Operation:

**Insertion**: Inserting all the nodes in the array of Nodes.

3. Binary Tree:

Importance: Using the Binary Tree we are *traversing* through the nodes and assigning the Huffman code to the characters.

Operation

**Traversal:** By Traversing the Tree are assigning Huffman Code to the characters. If we traverse to the left then we assign ‘0’ and if we traverse to the right then we assign ‘1’.

Algorithm:

1. Take Text File as an Input.

2. Calculate Frequency of each character in the file.

3. Create Nodes of characters with nonzero frequencies.

4. Insert the nodes in the minimum priority queue.

5. Remove two nodes from the minimum priority queue (left and right)🡪 add their frequencies and make a new node and link the left and the right nodes to the new node.🡪Insert the new node in the minimum priority queue.

6. Is there only one element left in the minimum priority queue?

No: Go to Step 5.

Yes: Go to Step 7.

7. Remove the node from the minimum priority queue.

8. Traverse through the node (Tree) and assign the Huffman code to the respective characters.

9. Create the new text file and write the respective Huffman codes according to the characters.

List of Program:

1. Node: For creating binary tree with left and right links and also it can store the character with its frequencies.
2. Min\_Priority\_Queue: For Sorting the Nodes according to their frequencies.
3. Huffman\_Tree:For Traversing and assigning the Huffman code to the respective characters.

References:

1. Data Structures and Algorithms in java, Second Edition, Robert Lafore, Pearson.
2. <https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffman_tutorial.html>