

# REPORT

IT-427

PROG ASSIGNMENT – 4

Full name – Shashi Teja P.V.

ULID – sparamk

SECRET DIRECTORY- “marvel”

Question- For this assignment, you will construct Huffman codes based on the given frequencies of 26 English alphabets in upper case plus the space character (it is hard coded in the given asg4.java). You should design a HuffmanCode class and put it in the same directory of asg4.java.

The commands to run the code for the given file

Compile – All the java files

Run – java asg4

### **Code Explanation-**

#### **Node.java**

I created a Node class which has various attributes

right Node, left Node , frequency, height, numberOfNodes, asciiValues

It has a parameterized constructor, which assigns values to it

#### **HuffmanCode.java**

In this I am having an parameterized constructor, most of the code is written in that only.

It takes an character array and frequencies array(Integer)

I am creating an priority queue which stores nodes in increasing order of the their frequencies

Creating the nodes and adding all the nodes into the queue

I am running a while loop until the queue size is greater than 1

In that, I am popping two nodes namely one and two and based on the given rules in pdf I am assigning them as either left or right node to a new node which I am creating. This gives us a **right-heavy** tree

And after that I am updating the newly created node data i.e

Frequencies, number of nodes, height and ascii value total.

I am collecting the final top root data and storing it.

### **printCodeWords()**

In this I am calling another method named **printCodeWordsAll(Node root, String s)**

### **printCodeWordsAll(Node root, String s)**

In this I am using recursion to traverse the tree and storing them in two maps which store the same data in reverse manner .

And I am printing them accordingly by traversing from root and its child nodes

### **String encode(String s)**

In this method I am reading the given string one by one character and encoding them with their code word.

### **String decode(String s)**

In this I am reading the encoded data one by one character and storing them in a string , once if the stored keyword matches with any value Then I am adding its original letter to the result string.

### **Data Structures Used-**

- Map- Hashmap to store the key and values of the letters
- Queue- to store the data based on frequencies i.e priority queue

Time and space complexity of my code – Big O ( $n \log n$ )

$n$  = number of letters in the given array

The sorting takes  $O(n \log n)$  – for priority queue.

And all other code takes  $O(n)$

So based on this my Time complexity is  $O(n \log n)$

Space complexity- Its  $O(n)$

My code takes  $O(n)$  to store the elements