import java.util.PriorityQueue;

import java.util.HashMap;

import java.util.Map;

class HuffmanNode implements Comparable<HuffmanNode> {

char data;

int frequency;

HuffmanNode left, right;

public HuffmanNode(char data, int frequency) {

this.data = data;

this.frequency = frequency;

}

public boolean isLeaf() {

return left == null && right == null;

}

@Override

public int compareTo(HuffmanNode other) {

return this.frequency - other.frequency;

}

}

public class DAA\_2 {

public static Map<Character, String> buildHuffmanTree(String text) {

// Count the frequency of each character

Map<Character, Integer> frequencies = new HashMap<>();

for (char c : text.toCharArray()) {

frequencies.put(c, frequencies.getOrDefault(c, 0) + 1);

}

// Create a priority queue for the Huffman nodes

PriorityQueue<HuffmanNode> pq = new PriorityQueue<>();

for (Map.Entry<Character, Integer> entry : frequencies.entrySet()) {

pq.add(new HuffmanNode(entry.getKey(), entry.getValue()));

}

// Build the Huffman tree

while (pq.size() > 1) {

HuffmanNode left = pq.poll();

HuffmanNode right = pq.poll();

HuffmanNode parent = new HuffmanNode('-', left.frequency + right.frequency);

parent.left = left;

parent.right = right;

pq.add(parent);

}

// Create a map to store the Huffman codes

Map<Character, String> huffmanCodes = new HashMap<>();

buildHuffmanCodes(pq.peek(), new StringBuilder(), huffmanCodes);

return huffmanCodes;

}

private static void buildHuffmanCodes(HuffmanNode node, StringBuilder code, Map<Character, String> huffmanCodes) {

if (node.isLeaf()) {

huffmanCodes.put(node.data, code.toString());

return;

}

if (node.left != null) {

code.append('0');

buildHuffmanCodes(node.left, code, huffmanCodes);

code.deleteCharAt(code.length() - 1);

}

if (node.right != null) {

code.append('1');

buildHuffmanCodes(node.right, code, huffmanCodes);

code.deleteCharAt(code.length() - 1);

}

}

public static void main(String[] args) {

String text = "this is an example for huffman encoding";

Map<Character, String> huffmanCodes = buildHuffmanTree(text);

System.out.println("Huffman Codes:");

for (Map.Entry<Character, String> entry : huffmanCodes.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}