TOPIC 5.7 : HUFFMAN CODING

Problem Statement  
Given a set of characters and their corresponding frequencies, construct the Huffman Tree and generate the Huffman Codes for each character.

Test Case 1  
Input:  
n = 4  
characters = ['a', 'b', 'c', 'd']  
frequencies = [5, 9, 12, 13]  
Output: [('a', '110'), ('b', '10'), ('c', '0'), ('d', '111')]

Test Case 2  
Input:  
n = 6  
characters = ['f', 'e', 'd', 'c', 'b', 'a']  
frequencies = [5, 9, 12, 13, 16, 45]  
Output: [('a', '0'), ('b', '101'), ('c', '100'), ('d', '111'), ('e', '1101'), ('f', '1100')]

Aim  
To write a program that constructs the Huffman Tree from given characters and frequencies, and generates Huffman Codes for each character.

Algorithm

1. Start
2. Create a priority queue with nodes as (frequency, character)
3. While more than one node remains:
   * Extract two nodes with smallest frequencies
   * Create a new internal node with these two as children and frequency equal to their sum
   * Insert the new node into the queue
4. The remaining node is the root of the Huffman Tree
5. Traverse the tree to assign binary codes: left edge → 0, right edge → 1
6. Store and return codes for each character
7. Stop

Input and Output  
A screenshot of a computer

AI-generated content may be incorrect.

Result  
The program successfully constructs the Huffman Tree and generates Huffman Codes for the given characters.

Performance Analysis  
Time Complexity: O(n log n) due to priority queue operations  
Space Complexity: O(n) for storing nodes and codes