Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 10

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. All edges along the path to a character contain a code digit. If they are on the left side of the tree, they will be a 0 (zero). If on the right, they'll be a 1 (one). Only the leaves will contain a letter and its frequency count. All other nodes will contain a null instead of a character, and the count of the frequency of all of it and its descendant characters.

Construct the Huffman tree for the following data and obtain its Huffman code.

Characters	**A**	**B**	**C**	**D**	**E**	**-**
Frequency/ Probability	**0.5**	**0.35**	**0.5**	**0.1**	**0.4**	**0.2**

(i) Encode text CAD-BE using the above code.

Input: CAD-BE

Output: 10011100110111100

(ii) Decode the text 1100110110 using the above information.

Input: 0011011100011100

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

Output: E-DAD

Code:-

```
from flask import Flask, render_template, request, redirect,
url_for, flash

import os



app = Flask(__name__)

app.secret_key = 'secret_key_for_flash_messages'



# Huffman Node Class

class Node:

    def __init__(self, char, freq, left=None, right=None):

        self.char = char

        self.freq = freq

        self.left = left

        self.right = right



# Build the Huffman Tree

def build_huffman_tree(char_freq):

    nodes = [Node(char, freq) for char, freq in
char_freq.items()]

    while len(nodes) > 1:
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
        nodes = sorted(nodes, key=lambda x: x.freq)
        left = nodes.pop(0)
        right = nodes.pop(0)
        merged = Node(None, left.freq + right.freq, left, right)
        nodes.append(merged)
    return nodes[0]


# Generate Huffman Codes
def generate_huffman_codes(root, current_code="", codes=None):
    if codes is None:
        codes = {}
    if root is None:
        return
    if root.char is not None:
        codes[root.char] = current_code
    generate_huffman_codes(root.left, current_code + "0", codes)
    generate_huffman_codes(root.right, current_code + "1", codes)
    return codes


# Encode Text
def encode(text, codes):
    try:
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
        return ''.join([codes[char] for char in text])
    except KeyError:
        return "Error: Invalid character in input."


# Decode Text
def decode(encoded_text, root):
    decoded_text = ""
    current_node = root
    for bit in encoded_text:
        current_node = current_node.left if bit == '0' else
current_node.right
        if current_node.char is not None:
            decoded_text += current_node.char
            current_node = root
    return decoded_text


@app.route('/', methods=['GET', 'POST'])
def index():
    huffman_codes = {}
    encoded_text = ""
    decoded_text = ""
    char_freq = {}
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
    if request.method == 'POST':

        try:

            # Read frequency input from the user

            freq_input =
request.form['char_freq'].strip().split(',')

            char_freq = {pair.split(':')[0].strip().upper():
float(pair.split(':')[1].strip()) for pair in freq_input}



            # Build the Huffman tree and generate Huffman codes

            huffman_tree = build_huffman_tree(char_freq)

            huffman_codes = generate_huffman_codes(huffman_tree)



            # Get action (encode/decode) and input text from the
user

            action = request.form['action']

            input_text =
request.form['input_text'].strip().upper()



            # Perform encoding or decoding

            if action == 'encode':

                encoded_text = encode(input_text, huffman_codes)

            elif action == 'decode':

                decoded_text = decode(input_text, huffman_tree)
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
        except Exception as e:
            flash(f"Error: {e}. Please enter valid input.")
            return redirect(url_for('index'))

    return render_template('index.html',
huffman_codes=huffman_codes,
                           encoded_text=encoded_text,
decoded_text=decoded_text)

if __name__ == '__main__':
    app.run(debug=True)
```

Html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Huffman Coding</title>
    <style>
        /* General Styles */
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
body {
    font-family: 'Arial', sans-serif;
    background-color: #f4f7fc;
    margin: 0;
    padding: 0;
    color: #333;
}
.container {
    width: 100%;
    max-width: 800px;
    margin: 50px auto;
    padding: 30px;
    background-color: white;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    font-size: 28px;
    margin-bottom: 20px;
    color: #4A90E2;
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
}


label {
    font-size: 16px;
    margin-bottom: 8px;
    display: block;
    color: #555;
}


input[type="text"],
select,
button {
    width: 100%;
    padding: 12px;
    font-size: 16px;
    margin-bottom: 20px;
    border-radius: 6px;
    border: 1px solid #ddd;
    box-sizing: border-box;
}


input[type="text"]:focus,
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
select:focus,

button:focus {

    outline: none;

    border-color: #4A90E2;

}


button {

    background-color: #4A90E2;

    color: white;

    cursor: pointer;

    border: none;

}


button:hover {

    background-color: #357ABD;

}


.flash {

    background-color: #ffb3b3;

    color: #ff0000;

    padding: 10px;

    margin-bottom: 20px;
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
        border-radius: 4px;
        border: 1px solid #ff0000;
    }

    .output h2 {
        font-size: 22px;
        margin-top: 30px;
        color: #333;
    }

    .output p {
        background-color: #f4f4f4;
        padding: 15px;
        border-radius: 6px;
        font-size: 18px;
        word-wrap: break-word;
    }

    .output ul {
        padding-left: 20px;
    }
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
.output ul li {
    font-size: 18px;
    margin: 5px 0;
}

.output {
    margin-top: 20px;
}

@media screen and (max-width: 600px) {
    .container {
        padding: 20px;
        margin-top: 20px;
    }

    input[type="text"],
    select,
    button {
        font-size: 14px;
        padding: 10px;
    }
}
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
    </style>
</head>
<body>
    <div class="container">
        <h1>Huffman Coding - Encode and Decode Text</h1>

        <!-- Flash Message for Errors -->
        {% with messages = get_flashed_messages() %}
        {% if messages %}
            <div class="flash">
                <ul>
                {% for message in messages %}
                    <li>{{ message }}</li>
                {% endfor %}
                </ul>
            </div>
        {% endif %}
        {% endwith %}

        <form method="POST">
            <div>
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
                <label for="char_freq">Character Frequencies
(comma-separated, e.g., A:0.5, B:0.3):</label>

                <input type="text" name="char_freq" required>

            </div>

            <div>

                <label for="input_text">Input Text:</label>

                <input type="text" name="input_text" required>

            </div>

            <div>

                <label for="action">Action
(Encode/Decode):</label>

                <select name="action">

                    <option value="encode">Encode</option>

                    <option value="decode">Decode</option>

                </select>

            </div>

            <div>

                <button type="submit">Submit</button>

            </div>

        </form>


        {% if huffman_codes %}

            <div class="output">
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
            <h2>Huffman Codes</h2>
            <ul>
                {% for char, code in huffman_codes.items() %}
                    <li>{{ char }}: {{ code }}</li>
                {% endfor %}
            </ul>
        </div>
    {% endif %}


    {% if encoded_text %}
        <div class="output">
            <h2>Encoded Text:</h2>
            <p>{{ encoded_text }}</p>
        </div>
    {% endif %}


    {% if decoded_text %}
        <div class="output">
            <h2>Decoded Text:</h2>
            <p>{{ decoded_text }}</p>
        </div>
    {% endif %}
```

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54

```
    </div>

</body>

</html>
```

Output

Name:-Devansh Koyani

Er no:- 22162171007

Batch:-54