15-211 : Fundamental Data Structures and Algorithms

SOLUTIONS
1. **Huffman compression.**

   (10) (a) Consider the following string:

   \texttt{she\_sells\_sea\_shells}

   and show the Huffman codeword for each character (left = 0, right = 1).

<table>
<thead>
<tr>
<th>char</th>
<th>freq</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>6</td>
<td>00</td>
</tr>
<tr>
<td>h</td>
<td>2</td>
<td>0111</td>
</tr>
<tr>
<td>e</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>3</td>
<td>010</td>
</tr>
<tr>
<td>l</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>0110</td>
</tr>
</tbody>
</table>

   **Solution:**

<table>
<thead>
<tr>
<th>char</th>
<th>freq</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>6</td>
<td>00</td>
</tr>
<tr>
<td>h</td>
<td>2</td>
<td>0111</td>
</tr>
<tr>
<td>e</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>3</td>
<td>010</td>
</tr>
<tr>
<td>l</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>0110</td>
</tr>
</tbody>
</table>

   (20) (b) How large will the compressed version of the above sentence be, in bits, including the header?

   **Solution:** For each character, multiply its frequency by the length, in bits, of its codeword. Then just add up these products to get 49 bits = 6 * 2 + 2 * 4 + 4 * 2 + 3 * 3 + 4 * 2 + 1 * 4.

   Header: 1 bit per node + 8 bits per char = 11 + 48 = 59.

   Total 49 + 59 = 108.

   (10) (c) Explain why the following codes

   \texttt{00, 01, 10, 110}

   cannot be Huffman codes for any probabilities?

   **Solution:** The Huffman code is optimal, The above one is not, because 00, 01, 10, 11 is clearly better.
2. LZW compression.

(a) Compress the following string with LZW:

\[ a \ b \ a \ b \ b \ a \ b \ b \ b \ b \]

The alphabet is \( \{a, b\} \) and the dictionary is initialized to:

\[ a : 0 \quad b : 1 \]

Show the final dictionary, as well as the compressed string.

**Solution:**

\[
\text{OUT: } 0 \ 1 \ 2 \ 3 \ 1 \ 6 \ 1 \\
a : 0 \quad b : 1 \quad ab : 2 \quad ba : 3 \quad abb : 4 \quad bab : 5 \\
bb : 6 \quad bbb : 7
\]

(b) Decompress the following string with LZW:

\[ 0 \ 1 \ 0 \ 3 \ 8 \ 7 \ 3 \ 4 \ 8 \]

The alphabet is \( \{a, b, c, d, e\} \) and the dictionary is initialized to:

\[ a : 0 \quad b : 1 \quad c : 2 \quad d : 3 \quad e : 4 \]

Show the final dictionary, as well as the decompressed string.

**Solution:**

\[
\text{OUT: } a \ b \ a \ d \ d \ a \ d \ d \ e \ d \ d \\
a : 0 \quad b : 1 \quad c : 2 \quad d : 3 \quad e : 4 \\
ab : 5 \quad ba : 6 \quad ad : 7 \quad dd : 8 \\
dda : 9 \quad add : 10 \quad de : 11 \quad ed : 12
\]
3. DFA. Draw a 3-state DFA that accepts the set of all bitstrings ending with 11.

Solution:

![DFA Diagram]

The DFA diagram shows a 3-state machine. The states are labeled 0, 1, and 1 (with an accept state marked by a double circle). The transitions are labeled 0 and 1, indicating the actions taken for each input symbol.