Huffman Coding—Asynchronous (Monday, October 4 / Tuesday, October 5)

Readings

- [Lecture Notes Chapter 15: Huffman Coding](#)

Problems

**Problem 1**

Construct an optimal Huffman coding for the following alphabet and frequency table:

<table>
<thead>
<tr>
<th>Character</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.4</td>
<td>0.3</td>
<td>0.15</td>
<td>0.1</td>
<td>0.05</td>
</tr>
</tbody>
</table>

What is the ABL, or average bits per letter, for this encoding?

**Solution**

The following tree $T$ would be produced:

```
  /\  \\
 A /  \\
 B /    \\
  C /    \\
   E    D
```

```
ABL(T) = \sum_{x \in S} f_x \cdot \text{depth}_T(x) = 0.4 \cdot 1 + 0.3 \cdot 2 + 0.15 \cdot 3 + 0.1 \cdot 4 + 0.05 \cdot 4 = 2.05
```

**Problem 2**

What are the pros and cons of Huffman Coding?

**Solution**

**Pros:** Huffman codings represent only the characters contained in the text, without wasting space in the encoding length for characters not present. The simplest method to encode a text of $n$ characters would be to use $\lceil \lg n \rceil$-bits, resulting in a fixed number of bits per character. Although we cannot reduce the number of bits used per encoding, Huffman optimizes this method because characters that occur more frequently take less bits to encode, minimizing the average bits per letter (ABL).

**Cons:** Since storing the symbol table takes space, Huffman codings may not be appropriate for small files where this size would be significant. Furthermore, encoding a text requires pre-processing to generate the table, which is ineffective given a data stream. Similarly, Huffman codings are not adaptive compression schemes. For example, given an alphabet with 4 characters and a text where the first half contains 2 characters equally frequently and the second half only contains the other 2 characters equally frequently, Huffman would assign 2 bits per letter. Instead, we could “adapt” by starting out with an encoding that represents just the 2 characters in the first half, which would only require 1 bit per character, and then change the encoding to represent just the 2 characters in the second half, which would again only require 1 bit per character. In other words, with some overhead, adaptive compression schemes could lead to an improvement over Huffman by essentially viewing the first half and second half as separate texts.
**Problem 3**

You have an alphabet with $n > 2$ letters and frequencies. You perform Huffman encoding on this alphabet, and notice that the character with the largest frequency is encoded by just a 0. In this alphabet, symbol $i$ occurs with probability $p_i; p_1 \geq p_2 \geq p_3 \geq \ldots \geq p_n$.

Given this alphabet and encoding, does there exist an assignment of probabilities to $p_1$ through $p_n$ such that $p_1 < \frac{1}{3}$? Justify your answer.

**Solution**

There does not exist an assignment of probabilities to $p_1$ through $p_n$ such that $p_1 < \frac{1}{3}$. Assume for the sake of contradiction that there exists an assignment such that $p_1 < \frac{1}{3}$. Consider the last step of the Huffman algorithm when our two final nodes are merged into one node. Let these two final nodes be $x$ and $y$. Because character 1, which has the highest frequency, has an encoding length of 1, it must have been merged via this step. WLOG, let $x$ be this character 1. Since this is the final step of Huffman, we know $p_x + p_y = 1$. From our assumption that $p_1 < \frac{1}{3}$, we know $p_y > \frac{2}{3}$.

Because $n > 2$, $y$ must be a node with at least 2 characters. Consider the time when $y$ was created. Let the nodes that were merged to become $y$ be $a$ and $b$. We know $p_a + p_b > \frac{2}{3}$, which implies that $\max\{p_a, p_b\} > \frac{1}{3}$. However, we have reached a contradiction. Huffman always merges the two smallest nodes, but when $y$ was created, node $x$ with $p_x < \frac{1}{3}$ was still available and unmerged. Hence, $x$ would have been chosen instead of $\max\{a, b\}$. Therefore, via contradiction, we have proved that the original claim is false and thus that there does not exist an assignment of probabilities as specified.