Solve the following problems before the exercise session and be prepared to present your solutions at the session.

1. Alice and Bob agree that she will choose a character uniformly at random from the string MISSISSIPPI and tells it to him. How much information (in bits) does she convey in the expected case? (This is called the entropy of the distribution of the characters and the 0th-order empirical entropy of the string.)
   SOLUTION: She conveys
   
   $$H(P) = (1/11) \log 11 + (4/11) \log(11/4) + (4/11) \log(11/4) + (2/11) \log(11/2) \approx 1.82$$
   
   bits of information in the expected case, where $P = 1/11, 4/11, 4/11, 2/11$.

2. Build a Shannon code for the probability distribution over the distinct characters $\{M, I, S, P\}$ according to which Alice chooses.
   SOLUTION: The codeword for $I$ is the first $\lceil \log(11/4) \rceil = 2$ bits of the binary representation of 0, i.e., 00; the codeword for $S$ is the first $\lceil \log(11/4) \rceil = 2$ bits of the binary representation of 4/11, i.e., 01; the codeword for $P$ is the first $\lceil \log(11/2) \rceil = 3$ bits of the binary representation of 8/11, i.e., 101; and the codeword for $M$ is the first $\lceil \log 11 \rceil = 4$ bits of the binary representation of 10/11, i.e., 1110.

3. Build a Huffman code for that probability distribution, make it canonical, and use it to encode MISSISSIPPI.
   SOLUTION: For example, the codeword for $I$ is 0, the codeword for $S$ is 10, the codeword for $M$ is 110 and the codeword for $P$ is 111, so the encoding of the whole string is 110010100101001111110.

4. What is the redundancy of your Huffman code (i.e., the amount by which the expected codeword length exceeds the entropy of the distribution)?
   The expected codeword length is $4/11 + 8/11 + 3/11 + 6/11 \approx 1.91$, so the redundancy is about 0.09.

5. Suppose we are given a sequence $W = w_1, \ldots, w_n$ of positive weights. Huffman’s algorithm builds a binary tree on $n$ leaves assigned those weights, whose weighted external path length (i.e., the sum over the leaves of their weights times their depths) is minimized. You saw Travis attempt at a proof of correctness in class; unscrambled the pieces below to obtain a coherent proof.

   (a) Without loss of generality, assume $w_1$ and $w_2$ are the smallest weights in $W$, so Huffman’s algorithm builds a tree $T$ for $W$ starting by replacing $w_1$ and $w_2$ with $w_1 + w_2$.

   (b) The difference between the weighted external path lengths of $T$ and $T'$ is $w_1 + w_2$.

   (c) Since difference between the weighted external path lengths of $T_{opt}$ and $T'_{opt}$ is $w_1 + w_2$, and $T'$ has minimum weighted external path length for $W'$, the difference between the weighted external path lengths of $T_{opt}$ and $T'$ is at least $w_1 + w_2$.

   (d) By induction. Huffman’s algorithm is trivially correct when given 1 weight. Assume Huffman’s algorithm is correct when given $n - 1$ weights.

---

1Travis forgot to write “expected” in a few places in the notes; he’s added it now.
By our inductive assumption, Huffman’s algorithm correctly builds an optimal tree $T'$ for $W' = w_1 + w_2, w_3, \ldots, w_n$.

Therefore, we need only show that the difference between the weighted external path lengths of an optimal tree for $W$ and $T'$ is at least $w_1 + w_2$.

As we proved in class, since $w_1$ and $w_2$ are the smallest weights in $W$, there exists an optimal tree $T_{opt}$ for $W$ in which the leaves with weights $w_1$ and $w_2$ are siblings.

If we remove the leaves with weights $w_1$ and $w_2$ from $T_{opt}$ and assign their parent weight $w_1 + w_2$, then we obtain a tree $T'_{opt}$ for $W'$.

**SOLUTION:** By induction. Huffman’s algorithm is trivially correct when given 1 weight. Assume Huffman’s algorithm is correct when given $n - 1$ weights. Without loss of generality, assume $w_1$ and $w_2$ are the smallest weights in $W$, so Huffman’s algorithm builds a tree $T$ for $W$ starting by replacing $w_1$ and $w_2$ with $w_1 + w_2$. By our inductive assumption, Huffman’s algorithm correctly builds an optimal tree $T'$ for $W' = w_1 + w_2, w_3, \ldots, w_n$. The difference between the weighted external path lengths of $T$ and $T'$ is $w_1 + w_2$. Therefore, we need only show that the difference between the weighted external path lengths of an optimal tree for $W$ and $T'$ is at least $w_1 + w_2$.

As we proved in class, since $w_1$ and $w_2$ are the smallest weights in $W$, there exists an optimal tree $T_{opt}$ for $W$ in which the leaves with weights $w_1$ and $w_2$ are siblings. If we remove the leaves with weights $w_1$ and $w_2$ from $T_{opt}$ and assign their parent weight $w_1 + w_2$, then we obtain a tree $T'_{opt}$ for $W'$. Since difference between the weighted external path lengths of $T_{opt}$ and $T'_{opt}$ is $w_1 + w_2$, and $T'$ has minimum weighted external path length for $W'$, the difference between the weighted external path lengths of $T_{opt}$ and $T'$ is at least $w_1 + w_2$. 