Huffman’s optimality proof:

An optimal binary prefix code must satisfy:

1. \( p(x_i) > p(x_j) \Rightarrow l_i \leq l_j \).
   
   \( l_i \) is the length of codeword \( i \).
   
   This is true because if it is not, we can swap the codewords then the new codebook would have smaller average code length which is a contradiction.

2. The two longest codewords have the same length. Else, chop a bit of the longer codeword; example:

   ![Diagram]

   - Two longest codewords: "c" and "d". Chop a bit of "d".

   Clearly, codebook \( C' \) has shorter average length than that of \( C \).
3. If two longest codewords differ only in the last bit, chop a bit off all of them.

Example: "b" and "e" have different bits other than the last bits:

- b → 1000
- e → 1111

"b" and "c" are the longest code words with the only the last bit is different.
$X = \{a, b, c, d, e\}: p_X = [0.25, 0.25, 0.2, 0.15, 0.17]$  

Huffman traceback given codes for progressively large Alphabet.

$P_2 = [0.55, 0.45], \ C_2 = [0, 1]$  
$P_3 = [0.25, 0.45, 0.3], \ C_3 = [00, 1, 01]$  
$P_4 = [0.25, 0.25, 0.2, 0.3], \ C_4 = [00, 10, 11, 01]$  
$P_5 = [0.25, 0.25, 0.2, 0.15, 0.15], \ C_5 = [00, 10, 11, 010, 011]$
Proof: Now, suppose one of these codes $\hat{c}_1, \hat{c}_2, \ldots$ cannot be further optimised.

- If $m > 2$ with $\hat{c}_m$ is the first sub-optimal code (note $\hat{c}_2$ is definitely optimal).
  
- An optimal $\hat{c}_m$ must have $L' \leq L_{\hat{c}_m}$
  
  $L_c$ is the average code length of $C_c$

- Rearrange the symbols with the longest code in $\hat{c}_m$ so that the two lowest probabilities $p_i$ and $p_j$ differ only in the last digit (doesn’t change the optimality).

- Merge $x_i$ and $x_j$ to create a new code $\hat{c}_{m-1}$ as in Huffman procedure.

- $L' = L_c - \delta i - \delta j$, since identical except $\hat{c}_{m-1} \leq \hat{c}_m$.
  
  $\delta i + \delta j$ shorter with probability $(p_i + p_j)$.

- But also $L' = L_{\hat{c}_{m-1}} - \delta i - \delta j$, hence $L' < L_{\hat{c}_{m-1}}$

which contradicts the assumption that $\hat{c}_m$ is the first sub-optimal code.