level zero

- has two sons
  - has none
Given $A, B, C, D$, Code into binary sequences.
1 1 0 0 1 0

C A B

\[
\begin{array}{c}
A \\
B \\
D \\
C \\
\end{array}
\]

0 1 0 1
Given a set of letters, how to code optimally?
Intuitively, we use short sequences to represent commonly used letters, and long sequences to represent rarely used letters like $Q$ and $Z$. 

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>40%</td>
<td>30%</td>
<td>20%</td>
<td>10%</td>
</tr>
</tbody>
</table>
Huffman’s Problem

Given a set of \( w_j \) with weight \( w_j \),

find a binary tree with

\[ \sum_i w_i l_i \] minimized,

where \( l_j \) is the level of the \( w_j \) in the tree.
\[ \sum_{i} w_i l_i = 19 \]
\[
\sum_{i} w_i l_i = 25
\]
A different way to calculate $\sum_i w_i l_i$.

The weight of a father = the weight of his two sons.

The sum of $n - 1$ fathers = $\sum_i w_i l_i$. 
Necessary condition for optimality.

Never give a large $w_i$ a lower level and a small $w_j$ a higher level.
Give the two smallest weights the same father.

\[ w_1 < w_2 < \cdots < w_n \]

Given \( w_1 \) and \( w_2 \) have the same father.