Ann, Bob, Charlie (*replace with your names*)
COSC 336
3/19/2020

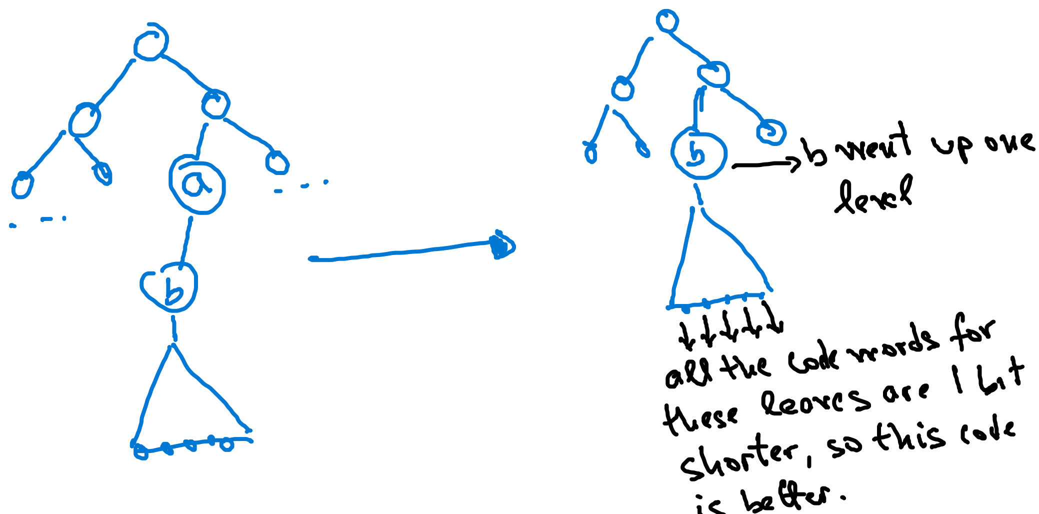
Assignment 5 - Solutions

Exercise 15.3-2, textbook, page 439

Prove that a non-full binary tree cannot correspond to an optimal prefix-free code. (note: A non-full binary tree is a binary tree that has a node which has only one child.)

Solution:

We show that a binary tree T (with corresponding prefix-free code C) that has a node with a single child, can be transformed into another binary tree T' (with corresponding prefix-free code C') so that $L(C') < L(C)$, and therefore C is not optimal (because C' is better). The transformation is simple: Suppose the node with a single node is a and its single child is b. Then we can delete a, which means that the node b is raised one level. This implies that the codewords in the subtree rooted at b, become one bit shorter, and therefore their contribution to the average length becomes smaller.



□

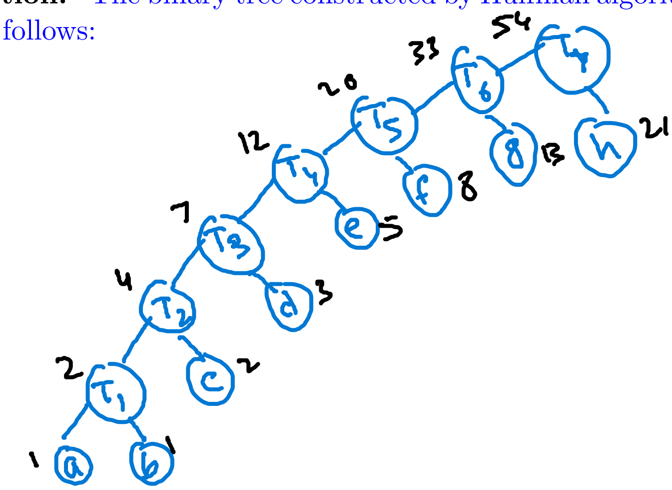
Exercise 15.3-3, textbook, page 439

What is an optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?

a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

Can you generalize your answer to find the optimal code when the frequencies are the first Fibonacci numbers?

Solution: The binary tree constructed by Huffman algorithm for the frequencies (1,1,2,3,5,8,13,21) is as follows:



So the codewords going in order from the highest frequency to the lowest frequency are: $1, 01, 0^21, 0^31, 0^41, 0^51, 0^61, 0^71$.

In general, if the frequencies are given by the first n Fibonacci numbers $F_1, F_2, \ldots, F_n$, the codewords of an optimal code are (again going from highest frequency to the lowest): $1, 01, 0^21, \ldots .0^{n-2}1, 0^n$.

The reason for that is that when we do Huffman algorithm, we never merge a frequency F_s with some higher frequency F_t (with $t > s$), because $F_s + F_t \geq F_s + F_{s+1} > F_s + F_{s-1} + \ldots + F_2 + F_1$. So, in Huffman algorithm, at each current moment we have the trees for $F_1 + F_2 + \ldots + F_s, F_{s+1}, F_{s+2}, \ldots, F_n$, and at this moment we merge $F_1 + F_2 + \ldots + F_s$, with F_{s+1}. This produces the codewords mentioned above. $\square$