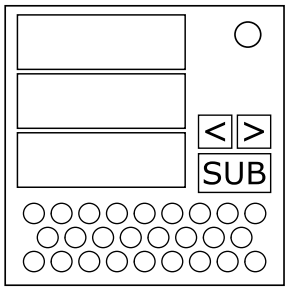
On the Subject of Red Huffman Ciphers



Ironically, more something for a Ravenclaw than a Hufflepuff.

This module contains 3 screens, a keyboard, 2 arrows, and a submit button that displays the current page you're on.

Pressing the left or right arrow takes you to the previous or next page. There are 2 pages.

To disarm the module, decrypt a word using the following three steps. Once you have the decrypted word, type it in using the keyboard. When you start typing, the screens go blank and the bottom screen will show what you are typing.

To clear your input, click one of the arrows.

Once you are satisfied with your input, press the button labeled "SUB" to submit your answer.

A	00000
B	00001
C	00010
D	00011
E	00100
F	00101
G	00110
H	00111
I	01000
J	01001
K	01010
L	01011
M	01100
N	01101
O	01110
P	01111
Q	10000
R	10001
S	10010
T	10011
U	1010
V	1011
W	1100
X	1101
Y	1110
Z	1111

Step 1: Encrypted Binary Retrieval

For this step, use the letters from the top, middle and bottom screens on page 1 and the top screen on page 2. Concatenate them in this order and convert this encoded string to binary by replacing each letter with a binary code from the table on the right.

Step 2: Huffman Tree Construction

Take the keyword from the bottom screen on page 2 and remove any duplicate letters. Take the entire alphabet and remove any letters that are in the keyword.

If the number of ports modulo 4 is 0 or 1, place the alphabet at the end of the keyword. Otherwise, place the alphabet at the beginning of the keyword. This is the alphabet key.

Now read the binary string from step 1 from left to right. Each 1 bit signifies to create a binary tree node, which will have two child nodes (called "left" and "right"). After each such node, decode its entire left subtree, then its entire right subtree. Each 0 bit signifies a leaf node, which will only contain a letter. As you encounter the leaf nodes, assign them the letters from the alphabet key in order. After decoding the tree structure, you will have some binary left over.

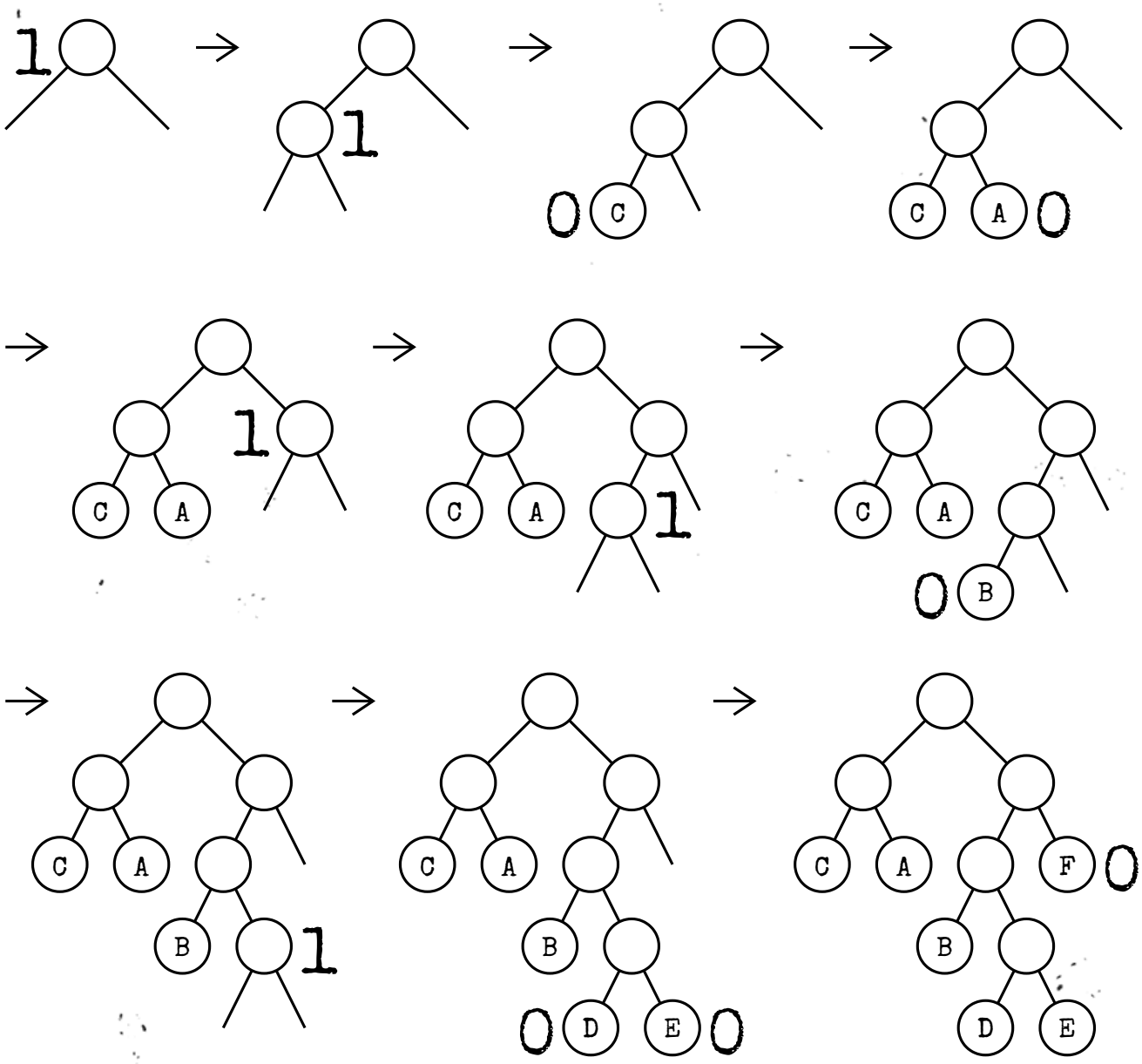
The next page illustrates these steps with an example.

Example for Step 1

Example code from screens: WXDIXL
W=1100 X=1101 D=00011 I=01000 X=1101 L=01011
Encrypted binary: 110011010001101000110101011

Example for Step 2

For brevity, we will pretend the alphabet has only 6 letters (A-F).
Keyword on last screen on page 2: CAB
Assuming no ports, alphabet key is: CABDEF



Leftover binary: 1101000110101011

Step 3: Huffman Decryption

Start at the root (top node) of the Huffman tree obtained in Step 1.

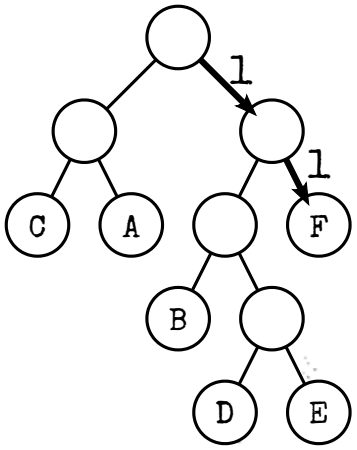
Read the first bit from the leftover binary. If it's a 0, move to the left child node, else the right child node.

Continue this until you reach a letter node. At this point, write down the letter, then move back to the root.

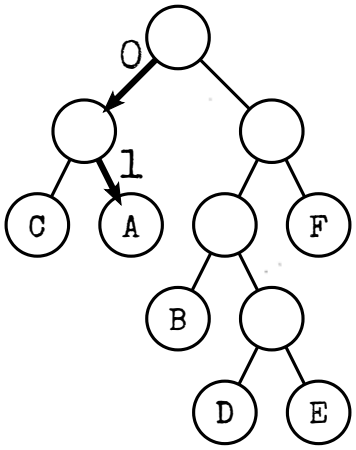
Repeat until all bits in the encrypted binary are exhausted. The received letters form the deciphered word.

Example for Step 3

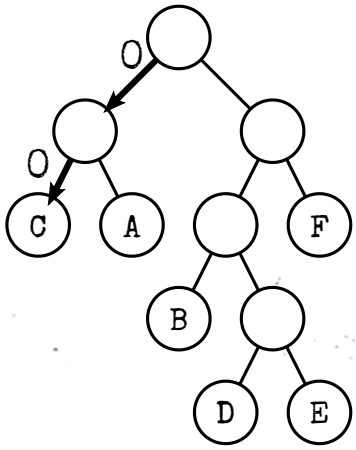
Leftover binary from Step 2: 1101000110101011 → deciphered word: FACADE



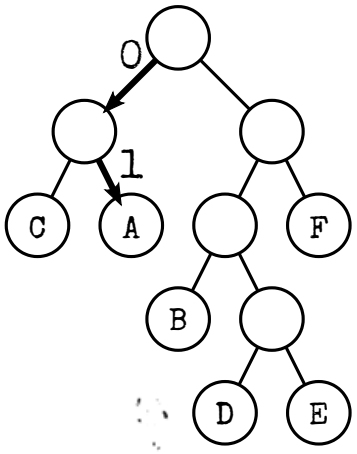
11 → F



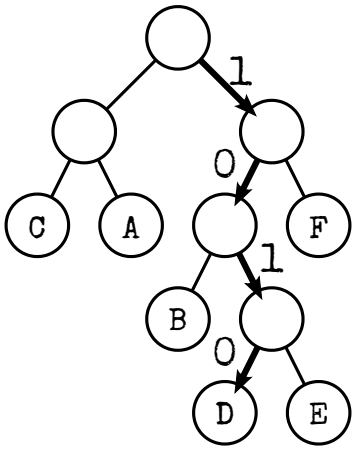
01 → A



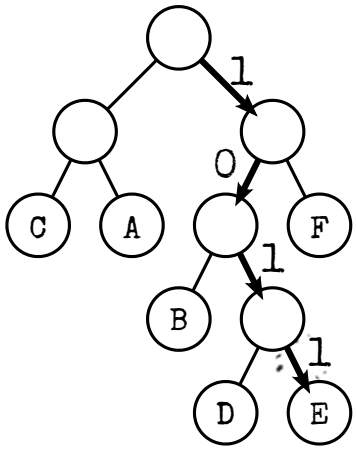
00 → C



01 → A



1010 → D



1011 → E