Written up for CS 360 lab 3, Dr. Plank Spring 2023
Written by UTA Andrew Mueller

Hello all,

The following is a more indepth visual of the tree that is made on the writeup for lab 3! If you were wondering how that tree was made, I am hoping this will guide you in writing your algorithm for this class. Admittedly the visual tree on the writeup did take me a good second to figure out when I took this class, so hopefully this will help you out if you are stuck on the first part! (There are major whitespace below here as Google Docs kinda sucks for this, but deal with it)

Color coding for the pictures is as follows:

Blue → Newly inserted string on a node
Purple → Newly created node
Red → NULL string

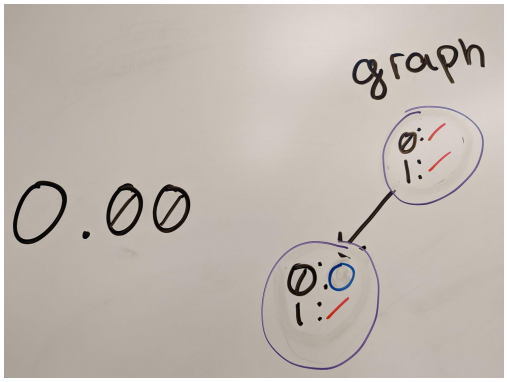
The code format file that is being used is on the writeup, but I will put it here for you convenience:

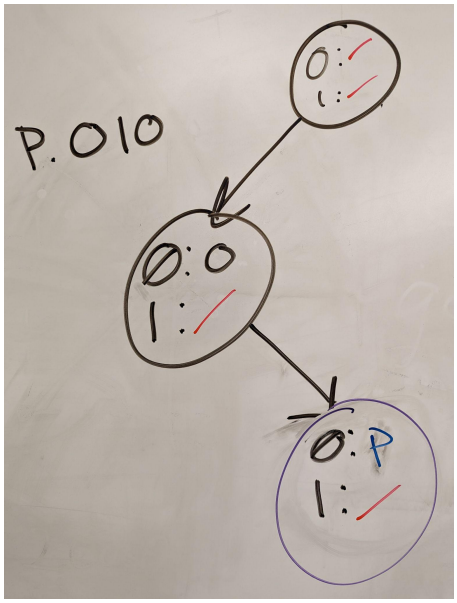
```
UNIX> xxd -g 1 data/t-code.txt
00000000: 6f 00 30 30 00 70 00 30 31 30 00 74 68 65 00 30  o.00.p.010.the.0
00000010: 31 31 00 20 00 31 30 00 0a 00 31 31 30 30 00 66 11. .10...1100.f
00000020: 00 31 31 30 31 00 74 00 31 31 31 00              .1101.t.111.
```

The full text is here without the color coding I did above!

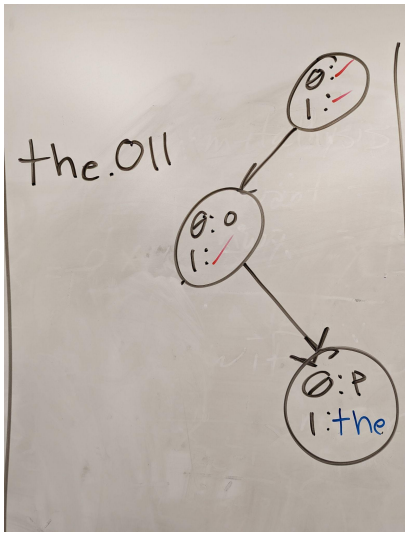
o.00.p.010.the.011. .10...1100.f.1101.t.111.

So let's get started here, for each picture, I have the string, and binary encoding that we are inserting at that moment. Hopefully going through this document fully will give you some idea on how you should go about inserting into the tree! Just a quick note, when I made this tree, the left child node will be the zero node, and the right one will be the 1 node.

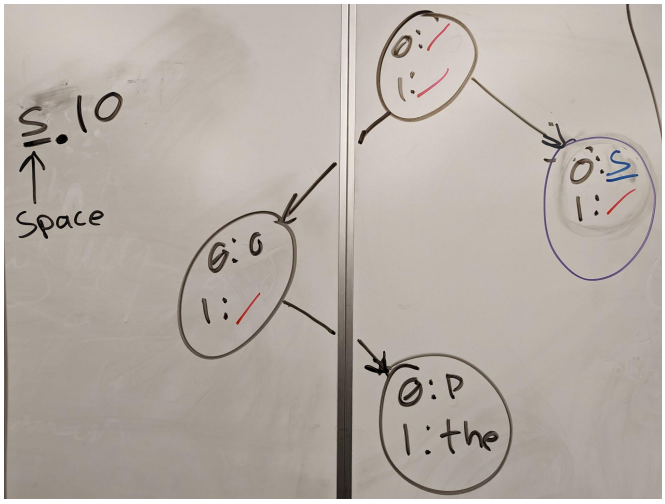
1. 

    a. As you can see, we have two purple nodes, indicating that we have to create them for inserting into this tree!
    b. As you can also see, the 0 string is set to “o” as the last bit in the binary encoding is 0
2. 

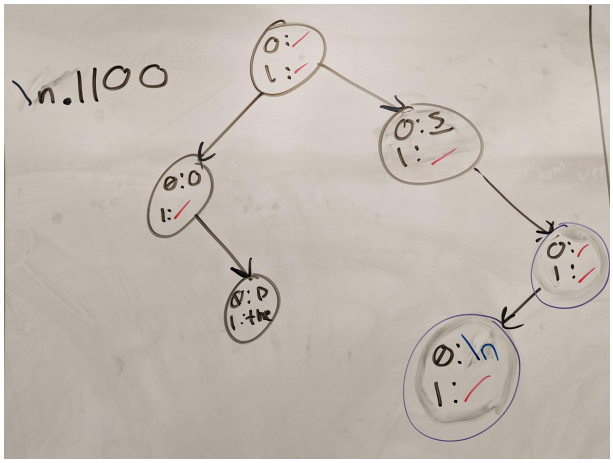
    a. As we can see here, we have a new purple node, and the 0 string on that node is set to “P”

3. 

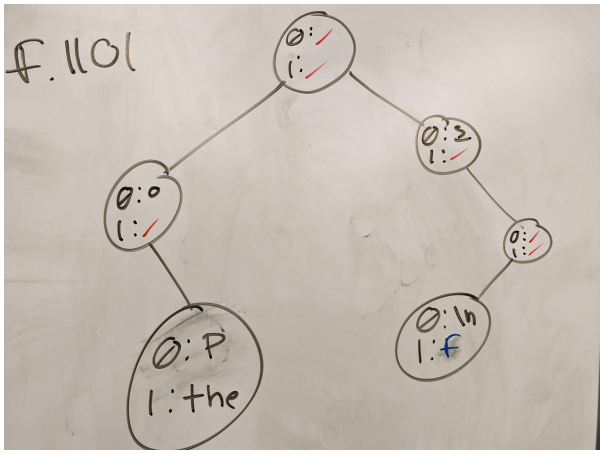
   a. We can see that there are NO purple nodes as the tree already had all the nodes we needed, however, for this string, we have set the 1 string to “the”

4. 

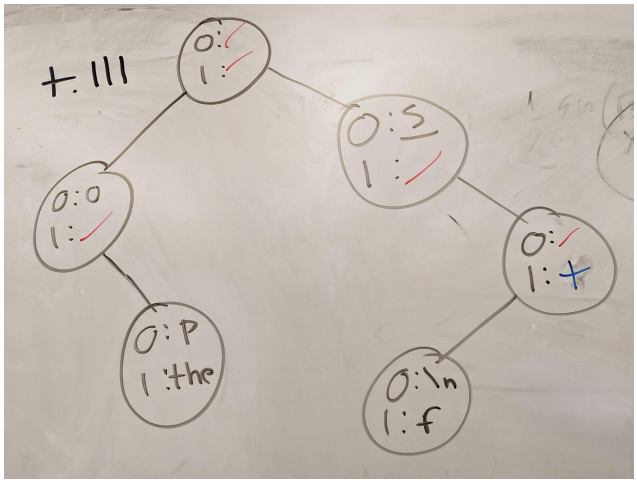
   a. Again, we have a new purple node, and the 0 string has been set for that node

5. 

   a. For this string, we needed to make two new nodes, and set the 0 string to the new line character

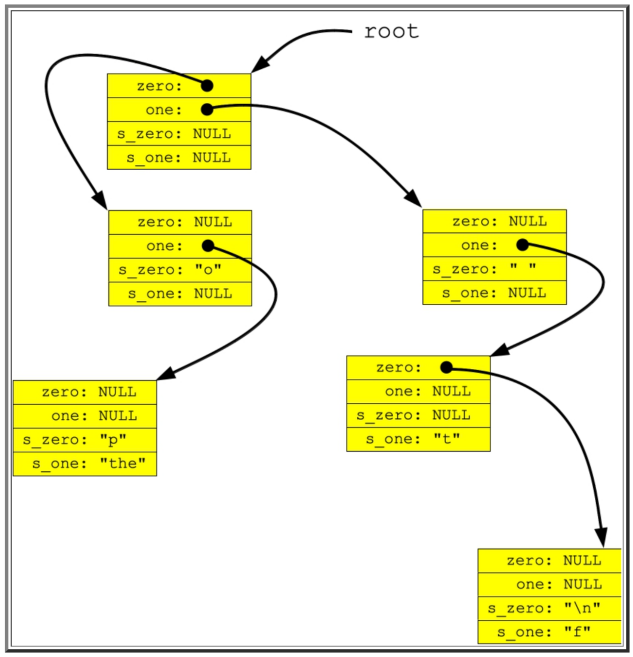
6. 

   a. Again here, we do not make any new nodes, but we do set the 1 string on the bottom right node to “f”

7. 

   a. Again, no new node, but we set the grandchildren of the root on the right side to “t” for the 1 string

The full tree on the writeup this looks like the last image above, but I will also put Dr. Planks image on here if that is a better visual for you:



Let us notice a couple things here:

1. Not every string is set, you will see a lot of NULL strings, and a lot of NULL pointers. You do not need to set anything that does not need to be set, and the tree is as minimal as needed for this string / binary encoding.
2. You do not need a sentinel node to point to the root, you can, but your logic just needs to change slightly.
3. You will ALWAYS start at the root of the tree when inserting new string / binary encodings!

Anyways, I hope this helps you walk through the logic on making the tree, and how you should go about doing this. I haven't given too much away as I know this is a major part of this lab, and you should know how to make a basic binary tree at this point!