Strategy for Range Functions

I couldn't figure out how to implement a proper strategy for the range functions so the functions for such functions were left blank.

Proof of Tree Height

```
int tmap_height(TMAP_PTR t){

   int temp;
   if(t->height == NULL)
      temp = (int)log2(t->pairs);
   else
      temp = t->height;

   return temp;
}
```

1. Claim: For $n > 0$, prove by induction that the height of the tree is O(1) on n
2. Base Case: n = 1; T(1) = 1
3. Induction Hypothesis: Assume that for arbitrary n, $T(n) <= n$
4. $T(n+1) = T(n) + 1$
   $<= n + 1$ by Induction Hypothesis
5. Since the inner statements are always constant, the function is O(1)

Bonus Operations

I was unable to complete the bonus operations and left them blank.

Procedure

To test the code, I had to use some test functions borrowed from exercises with binary trees during lab. Since we're dealing with arrays and nested structs, I had to first use code that didn't involve these nested structs. Had this code been used for this project, it would've been very lengthy (and probably wouldn't fit accordingly to runtime), but it was a functional program. Eventually, I was able to implement the nested structs and pointers that were given in the header file.

The sanity check that I had to run included a main in the tmap.c file (which was later deleted) that included test numbers in an array. Since some functions were missing, there were some issues with running the program with some of these functions. The main included some arbitrary numbers in an array and print statements for each function called into main (to check that it was traversing the binary tree and balancing it).

By creating a test main in the tmap.c file, I was able to run checks with names and numbers. I included extremely large numbers (such as 20.5) and very small numbers (like 0.99). While it worked with my extremely lengthy code that I used during labs, I somehow managed to make it work for this project. Since I couldn't get the range functions to properly work, there has been some issues with this.