

COP 3502 Summer 20 Section 2 Recitation Program #4

Cheating a Boolean Tree

For each recitation program, in order to get full credit, you must submit your solution to open.kattis.com and get your solution accepted on all test cases. In addition, each one will have some separate requirements to submit. When submitting your work to Webcourses, please carefully read the corresponding directions document before submitting all of your files.

NOTE: Over the course of the semester, you must submit TWO out of the four recitation programs. It is expected that while you are in recitation, you start working on each of them. But, afterwards, you can choose which two to finish up.

What This Program Is Testing

This is a "tree" program, but the input is always a complete binary tree, thus, just like a heap, the tree can be stored in an array. Since both storage techniques are taught in the course, you may use either.

What this program is testing is creating an algorithm on a tree. Typically, we must make a recursive call on a subtree (left and/or right), but in this program there is one other parameter we must pass in: whether we want to make the node false (0) or we want to make the node true (1). Due to the syntax of C, other items will be passed into the recursive function, but the two that matter are the node and which value to set the node. The result to the query can be determined from at most four recursive calls to the child nodes.

Run-Time Question

It may appear that this algorithm will receive a Time Limit Exceeded on a tree with 10,000 nodes, as the run time might appear to be exponential (multiple recursive calls like Towers of Hanoi). Explain, informally, why the algorithm, which may make upto four recursive calls, runs fast enough.

What to Submit

Please submit the following:

- 1) Your source file, booleantree.c, that was Accepted.
- 2) A screenshot of your solution's "Accepted" status on Kattis.
- 3) A document (.txt, .doc or .docx) explaining WHY your algorithm runs fast enough.