1. Describe and analyze an algorithm that rebuilds an arbitrary n-node binary search tree T, some of whose nodes may be marked "dead", into a perfectly balanced binary search tree, in $O(n)$ time. The output tree should contain each of the unmarked nodes in T and nothing else.

   For full credit, your algorithm should use only $O(\log n)$ space, *including the call stack* but not including the input tree T. *[Hint: Use rotations to reorganize the tree.]* For extra credit, describe an algorithm that uses only $O(1)$ additional space. *[Hint: No recursion!]*

The remaining problems are for you play with on your own.
Discussion in office hours or on Discord is welcome, but don't submit solutions!

---

2. [*Open Data Structures* Exercise 8.11]

   Design and implement a data structure that maintains a sequence of items subject to the following operations. (The inputs to all operations are pointers/references directly to sequence items.)

   - InsertAfter(x): Create a new item, insert it immediately x in the sequence (or at the start of the sequence if x is null), and return a pointer/reference to that new item.
   - Delete(x): Delete x from the sequence.
   - Before?(x, y): Return True if x comes before y in the sequence, and False otherwise.

   InsertAfter and Delete should each run in $O(\log n)$ amortized time, and Before? should run in $O(1)$ worst-case time.

   This data structure can be implemented by storing the elements in something like a scapegoat tree, in the same order that they occur in the sequence. To implement Before in constant time, each element x is labelled with an integer that encodes the path from the root to x. Then Before?(x, y) can be implemented by comparing the labels of x and y. *[Hint: We can reasonably compare two $O(\log n)$-bit integers in $O(1)$ time.]*

   [Hint: You can also think of this as the "BASIC Line Numbering Problem". In early versions of the programming language BASIC, every line of code must have a unique line number. *There were no editors; the only way to add or change a line of code was to type the entire line, starting with its chosen number. Renumbering 25 lines of code required retyping 25 lines of code (and possibly typing* `DELETE <num>` *25 times to get rid of the old lines). So it was recommended practice to use multiples of 10 as initial line numbers, to leave space for intermediate lines. And of course, no sane programmer writes their code in order from first line to last.]*