Assignment 3: Searching!

Due: Oct 13, 4:30 PM

Note: Absolutely no late submissions will be accepted.

This assignment is designed to demonstrate to you the benefit of using more advanced structures such as balanced trees for searching.

You are required to write three programs, each of which has the same function, but a different underlying algorithm. The programs take as input a file, containing a list of integers (each of these entries is seperated by a newline). The program is required to implement methods `insert(k)`, `find(k)`, and `delete(k)`. `find(k)` returns a pointer to the node that contains `k`, if it exists, otherwise, it returns `NULL`. `delete(k)` deletes *all* instances of `k` from the list.

The three programs implement this functionality in the following three ways:

- The first program uses a sorted list (an array) of the numbers, search relies on binary search.
- The second program uses a binary search tree (not necessarily balanced).
- The third program builds an AVL tree from an input list.

Notes: (i) do not use arrays to implement trees, (ii) make sure you handle duplicates appropriately.

Implement these three algorithms in three different programs. Test these programs on the input list from `http://www.cs.purdue.edu/homes/ayg/CS251/code/data_search`. This list contains 100,000 entries. In this list, search the entries in the following file and delete them `http://www.cs.purdue.edu/homes/ayg/CS251/code/search_list` (there are 1000 entries in this file).

Note the time for each of the program executions and include the timings as a separate file with your code for submission.