Lab 4

Due: at the end of class

1 Reading

Skim Chapters 12.1 – 12.3 of the textbook.

2 Exercises

Write out solutions to the exercises below, following the structure of the textbook. Please include the names of everyone in your group.

1. No matter the approach, every comparison-based sorting algorithm takes $n \lg n$ steps with some inputs. Argue that, as a consequence of this, any comparison-based algorithm for constructing a binary search tree from an unsorted list of n elements takes $\Omega(n \lg n)$ time, no matter what order the elements are given.

   (Hint: Consider converting a binary search tree to a sorted list.)

2. Consider a (false) property of binary search trees: Suppose that the search for key k in a binary search tree ends up **in a leaf**. Consider three sets: A, the keys to the left of the search path; B, the keys on the search path; and C, the keys to the right of the search path.

   The property claims that any keys $a \in A$ and $b \in B$ must satisfy $a \leq b$, and any keys $b \in B$ and $c \in C$ must satisfy $b \leq c$. Give a counterexample to the property with the *fewest number of nodes*. Draw the tree and mark k and the sets.

3. State the worst-case runtime for the following operations, assuming that there are n elements in the data structure:

   Inserting a new value into an unsorted array (assuming there is space):

   Inserting a new value into a sorted array (assuming there is space):

   Finding a value in an unsorted array:

   Finding a value in a sorted array:

   Inserting a new value into an unsorted (double) linked list:

   Finding a value in a sorted linked list:

   Inserting a new value into an array-backed heap (assuming there is space):

3 Grading

Exercise 1: 30%

Exercise 2: 35%

Exercise 3: 35%