(Add/Remove operations)

Root is going to change in a recursive call for the subtree (recRemove)

Removing a leaf node – easiest case to remove the bottom of the tree. (Just set it to null)

Set link of the parent to null,

Don’t set the object to null, because that will not have the desired effect (which is garbage)

Removing a node with 1 child – also easy parent needs to skip the node to the child.

Go to the parent and tell it to

Removing an internal node somewhere in the complicated because of Order of the subtree

The solution is pretty simple, even though thinking about it could be really difficult.

Going through the left link and the largest value, OR the right tree with the smallest value

(Remaining Tree operations) read in the textbook

Types of variations of trees

Very complex applications like in Motherboard circuitry in a logic gates implementation

Or a simpler Decision tree; Spam filter e.g. (Ask a series of questions and the tree stores the answers).

Geo applications like with something called a R data set???

Many examples of machine learning (languages, and language processing)...tries, which is a special word for the field of natural language processing and is pronounced like tree, but it comes from retrieves.

Balanced searches and algorithms.

Hashcodes without maps – hasing without maps can be useful as well.

For testable materials

Adding and removing conceptually

Special cases for removal (psuedo-code)

Tail recursion???