

Tutorial 6, Week 7 - Binary Search Trees

1 Before Your Tutorial

- Skim the lecture section about Binary Search Trees (or any similar resource really).
- Remember the rule for how Binary Search Trees work.

2 During Your Tutorial

- Complete the code for a Binary Search Tree using the `BinarySearchTree.h` and `BinarySearchTree.cpp` as templates.
 - Note that not all parts of this are as easy as others.
 - Start with `insert(int)` and `contains(int)`.
 - Erasing something from the tree requires slightly different behaviour for different cases. Sketch some trees and try to design an algorithm to achieve this.
 - For the `valuesInOrder()` function, look at your tree sketches. where is the smallest element? Where is the largest? What order do you have to visit the vertices to visit them in order?

3 Extensions

- Extend the BST to something a bit more useful by separating the role of key and value (at the moment the key is the value):
 - Try first by using `ints` as keys to store `string` values.
 - If you're happy with that, template it up! You can have a different template type for the key and the value.

- Consider how you might generate usable keys from the values, so the user don't have to specify the key, but you get keys with nice properties (think about the lecture you just saw).
- For those who have waaay too much time on their hands, research AVL Trees, and have a go at extending your BST to an AVL Tree (this is not trivial to do, but not so hard).