Binary Search Trees:

See the chapter 12 in [CLRS] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd Edition). MIT Press and McGraw-Hill. 2009.

AVL Trees:

See the chapters 5.11.1, 5.11.2 [here](http://www.cs.yale.edu/homes/aspnes/classes/223/notes.pdf).

<https://en.wikipedia.org/wiki/AVL_tree>

See this [visualization](https://www.cs.usfca.edu/~galles/visualization/AVLtree.html). Play with this AVL tree by adding and deleting elements to see how it manages to keep being balanced.

BST Applications:

See the chapters 14.1, 14.2 in [CLRS] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd Edition). MIT Press and McGraw-Hill. 2009.

Splay Trees:

See the chapter 5.11.6 [here](http://www.cs.yale.edu/homes/aspnes/classes/223/notes.pdf).

Also see this [visualization](http://www.cs.usfca.edu/~galles/visualization/SplayTree.html). Play with it by adding and erasing keys from it, and see how it can be unbalanced, in contrast with [AVL tree](http://www.cs.usfca.edu/~galles/visualization/AVLtree.html), but pulls the keys it works with to the top.

Also see this [answer](http://stackoverflow.com/questions/7467079/difference-between-avl-trees-and-splay-trees) about comparison of AVL trees and Splay trees.

Also see the original [paper](https://www.cs.cmu.edu/~sleator/papers/self-adjusting.pdf) on Splay trees.