Boris Topalov, bnt4yb, 2/26/2020, 5pm lab

1. The outputs of our AVL Trees when printing the path to our target nodes were shorter than the outputs when we were using Binary Search Trees. This means that our AVL Trees traveled to less nodes than the Binary Search Trees, so they found the path to our target node faster. So AVL Trees perform better on average.
2. BSTs are easier to implement since they don't have to keep the tree balanced and update the heights of each subtree whenever we insert a new element into the tree, unlike AVL Trees. The trade-off with this is that if we have a really large tree, it could be very unbalanced and it would take a really long time to find the path to a node, but with AVL trees it would be balanced and would take much less time.
3. When we have really large inputs we would probably prefer to use AVL Trees because the runtime, on average, of its find will be log(n). So if we are looking things up a lot we would want to use AVL Trees since they are much faster at finding than BSTs.