Refactoring:

* Is there a better data structure to use to store your data?

Probably the Binary Search Tree is a bit over the top, and a linked list could have been used.

* Would a different searching algorithm improve efficiency?

I can’t think of a better searching algorithm. The time efficiency for a perfectly balanced BST is O(logN).

It may be noted that in the current running program it’s not a perfectly balanced tree, because one node at depth 3 is empty. In our sample we have 40 elements. Those could be fit into a BST of depth 6. However our created BST has a single node at depth 10, 3 nodes at depth 9 and 5 nodes at depth 8, on the right side of the tree. The left part of the tree only reaches depth 7. So, it’s not perfectly balanced, but it has some improved efficiency, instead of making 40 comparisons, it makes an average of 7-8, and in some specific cases 9-10.

So in conclusion, yes there’s a margin to improve efficiency, but we can consider that is good enough as it stands.

* Is there documentation in your program to make your code readable?

The code is plagued with comments to explain the different functions and flow.