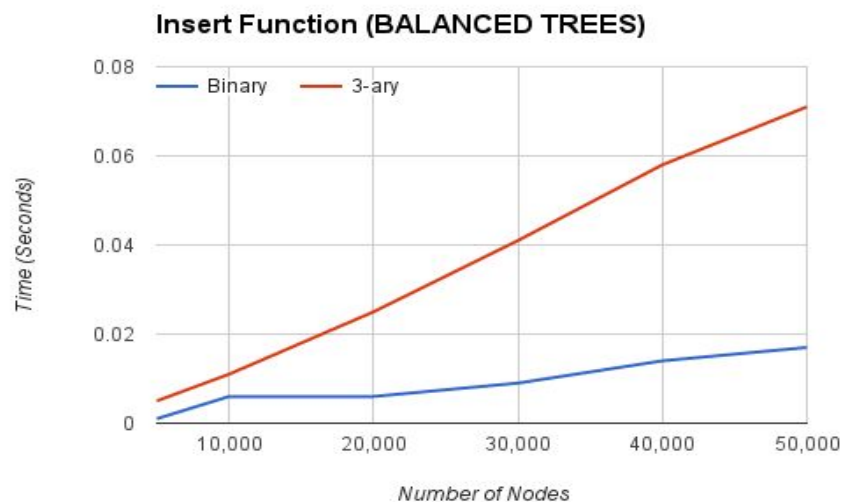


Project 2

Project 2 examines the difference between runtimes for a binary search tree and 3-ary search trees. The runtime complexity for both trees is $O(\log N)$. However, even though both trees scale logarithmically with input, the actual functions behind the trees shows the 3-ary tree functions resemble a $\log_3 N$ time, while the binary tree functions are $\log_2 N$ time. This should result in a slower expected time to complete functions for the 3-ary tree than for the binary tree for large inputs.

Multiple tests were run checking for the runtime of insert, remove, and find for balanced/unbalanced binary trees and balanced/unbalanced 3-ary trees. The unbalanced tree functions were magnitudes slower than their balanced counterparts. As for comparison between the two balanced trees, runtimes for the insert function can be seen below in the graph. Also tested were the remove and find functions, which yielded similar results.



As expected, the BST functions are faster than 3ST functions because it is easier to traverse nodes with 2 children than ones with 3 children. This is a reflection of the $\log_3 N$ and the $\log_2 N$ traversal times.