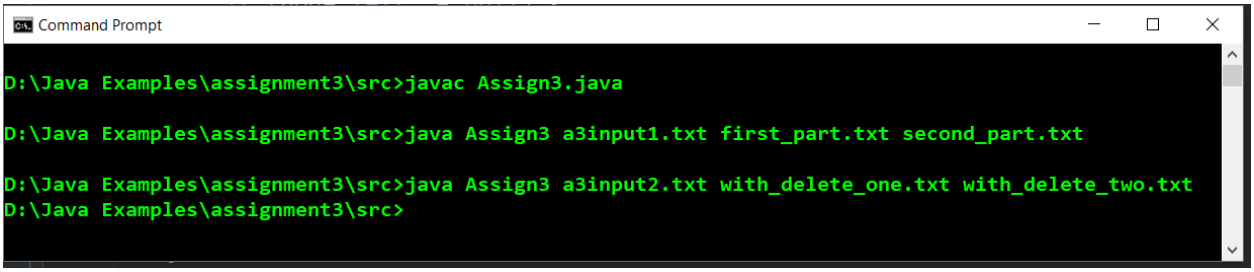
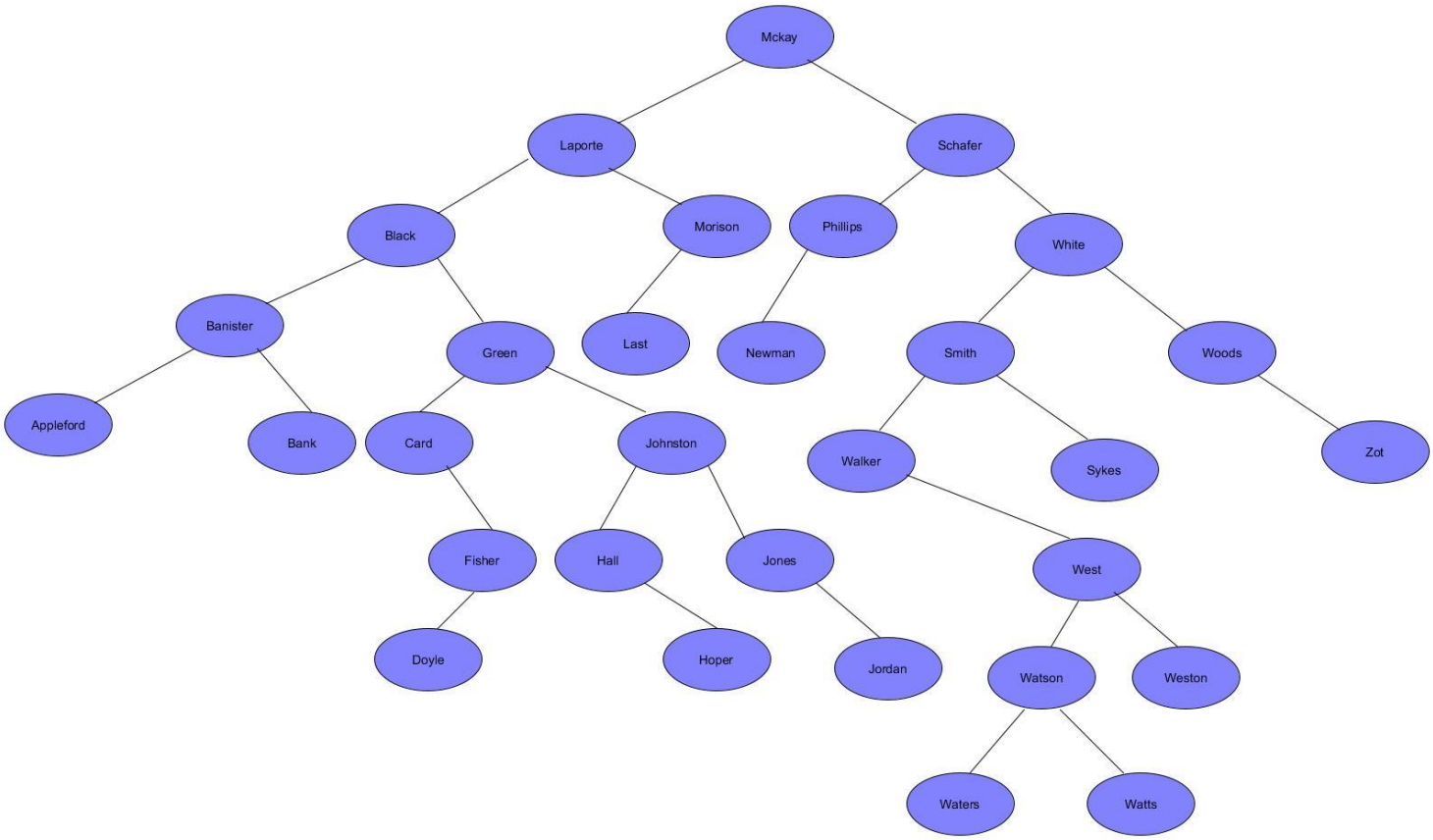
1. How to run:



The resulting files are provided as well.



Q1.

Assuming that the records are inserted into the tree in random order, what is the height of your tree expressed using big-O notation?

For a balanced tree with n nodes: • height = O(log2n).

2. What is the worst-case height of the tree? What input gives the worst case?

With a binary search tree, there's no way to ensure that the tree remains balanced. • can degenerate to O(n) time. Worst case occurs for skewed tree and worst case height becomes O(n).

What is the worst-case space complexity of the depth-first, in-order traversal and breadth-first traversal? Compare your implementation of these two methods: is there one that will outperform another in terms of memory usage for a specific data set?

All four traversals require O(n) time as they visit every node exactly once.

It is evident that extra space required for Level order traversal is likely to be more when tree is more balanced and extra space for Depth First Traversal is likely to be more when tree is less balanced.Breadth-first search for binary tree array is trivial. Since elements in binary tree array are placed level by level and bfs is also checking element level by level, bfs for binary tree array is just linear search. Obviously, BFS on array wins. It’s just a linear search