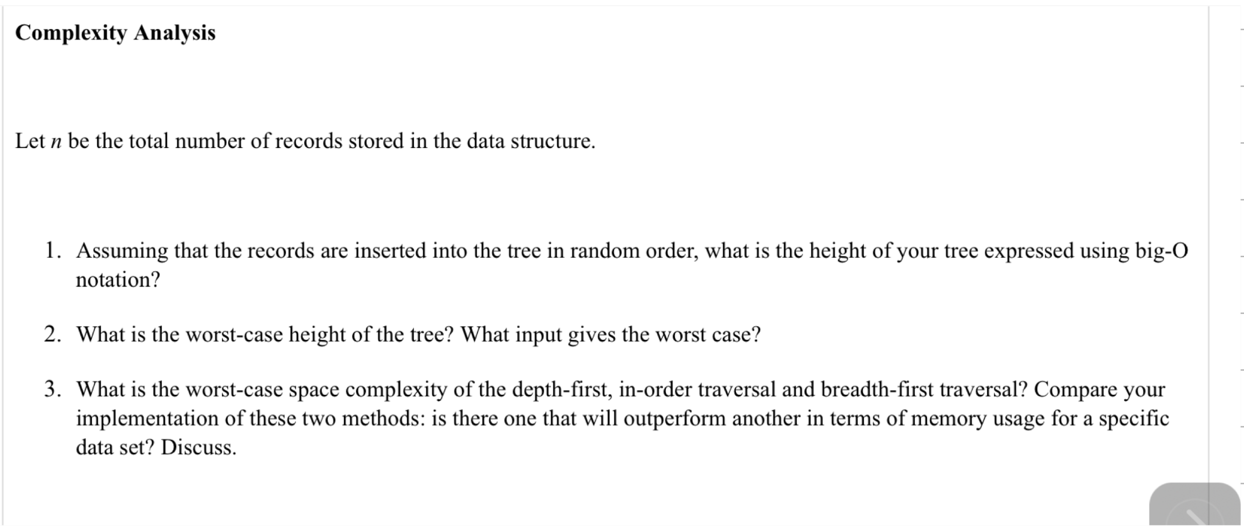
Name: Md SHaherier Khan

UCID: 30132195

TA: Xi Wang

Report Assignment 3



Complexity Analysis

Let n be the total number of records stored in the data structure.

1. Assuming that the records are inserted into the tree in random order, what is the height of your tree expressed using big-O notation?
2. What is the worst-case height of the tree? What input gives the worst case?
3. 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? Discuss.

(1) Following the rules of binary tree we can insert in the left of tree if data is less than or equal to parent node., else it would be the right.
Suppose the binary tree is balanced so it has a node can have a max of 2 children.
Suppose k is the height of root to longest leaf node . so the max number of node is 2^k , k also represent the the level. $2^0 = 1$ (root), so to find $k = \log_2(n)$. The big O notation is $O(\lg n)$.

(2) The worst case for a tree is when it degenerates to a linklist causing the data to be skewed to the left or right. So that the $O(n)$ as the height become $n-1$
The data that would cause such a linklist would be if data is sort

(3) Space complexity is worst case for DFS when max height is $n-1$ where n is the number of nodes
This $O(n)$ as big notation. height is $(n-1)$ when tree degeates to $n-1$

Space compexity is worst case for BFS when max width of tree is 2^k where k is max height of tree
the more complete / balance the tree the more width increase so does the space complexity. Also more balance indicate more node are stored in quee. Since max node is 2^k in quee stores $k = \log(n)$. , k is the high
so complexity $O(n)$.

To summarize

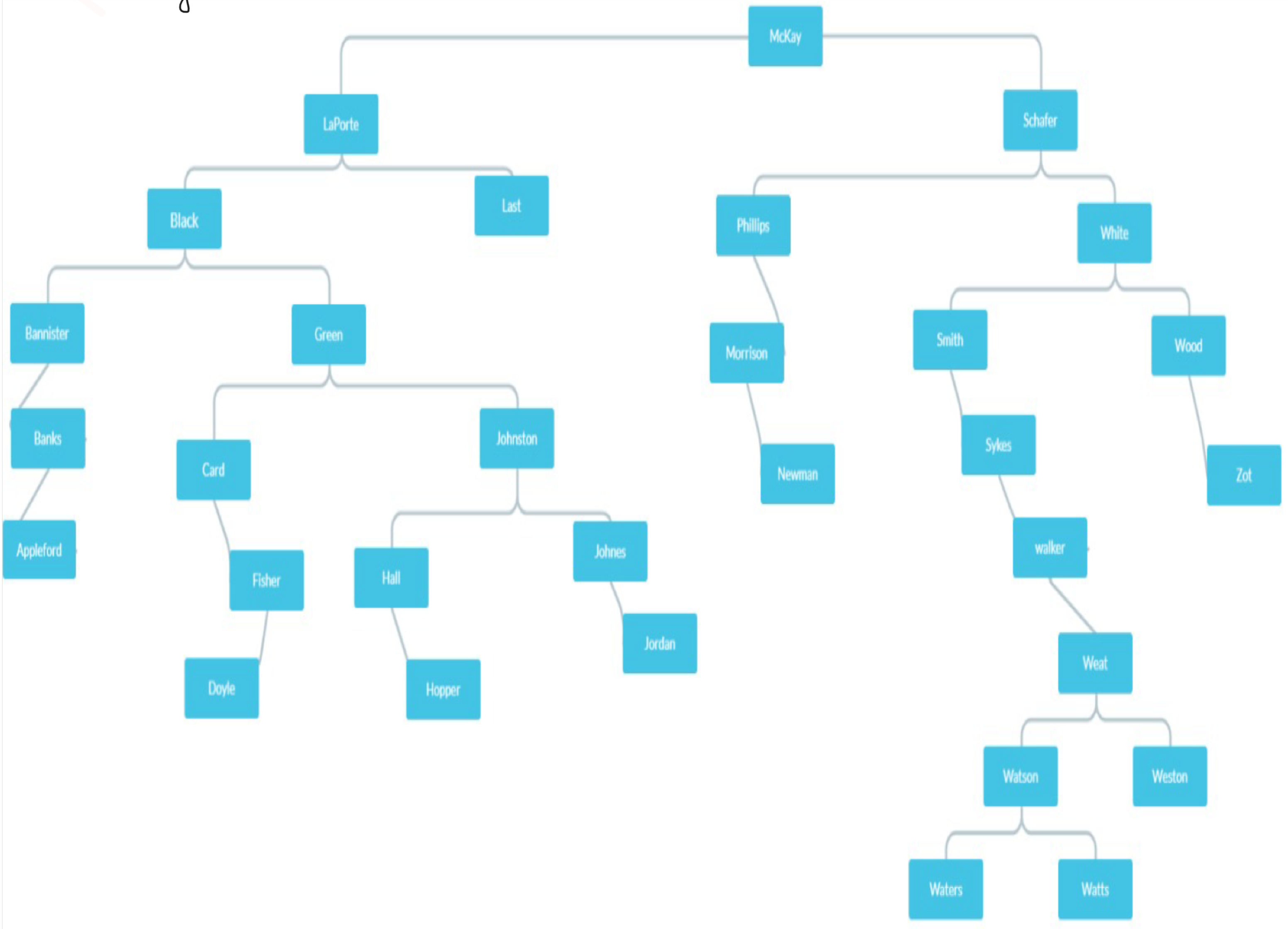
If the tree is well balances, it better to use DFS than BFS.

If the tree degerates to linklist, it better to use BFS than DFS

If the Node that we are looking for is at the top then use BFS

If the node that we are looking for is at the bottom then DFS

Tree diagram:


McKay
LaPorte
Schafer
Black
Last
Phillips
White
Bannister
Green
Morrison
Smith
Wood
Banks
Card
Johnston
Newman
Sykes
Zot
Appleford
Fisher
Hall
Johnes
walker
Jordan
Doyle
Hopper
Weat
Watson
Weston
Waters
Watts