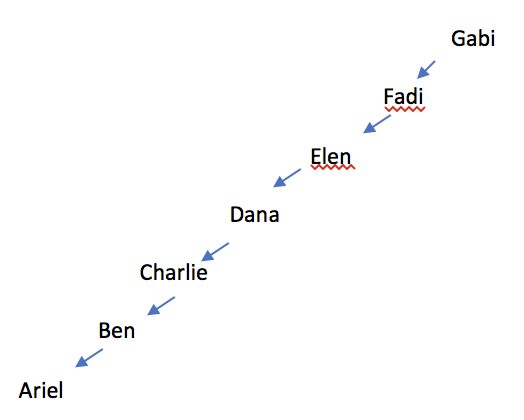
**Critical Thinking**

**Question 1:**

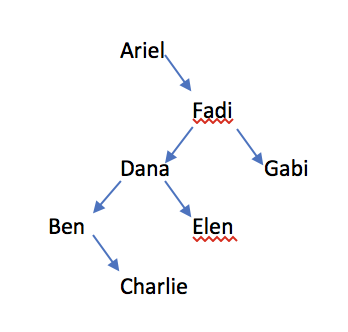
Insert seven people into the tree in alphabetical order and observe the tree structure by printing the tree string. What do you notice about the structure of the tree? What does this tell us about the worst case runtime for a single operation (i.e. search, insert, delete) in a splay tree? How many steps would it take if you inserted another element, following the alphabetical order from before?

Started by inserting Ariel….Gabi



* The tree is reversed and it is one side heavy. It will look like a linked list.
* If I insert another element it will take 1 steps to insert and n steps to splay.
* The worst case for a single operation will be O(n).

Now do a search for the first name you added into the tree, and print out the new tree structure. Has the tree changed significantly? What is the depth of the tree now? What if you insert nine people in alphabetical order to begin with? Or 11 people?



* The tree has a smaller depth (4)
* If we inserted more people to begin with we would have a fuller tree after the search.

**Question 2:**

Now imagine that we modified splay, got rid of the zig-zig and the zag-zag steps and only did single rotations during our new splay-like operation (if x is a right child rotate left and vice versa) until the node is at the top. What would happen if you did this splay- like operation on the leaf of this tree to bring G to the top:

