

Part A:

1. Write `rotateWithLeftChild`, which takes an `avlNode`, and rotate with its Left Child, update the height and return the new node.
2. Write a function called `printByNumChars` to print all given [input](#) names in the list, grouped by number of characters. The runtime must be $O(n \log n)$ where n is the number of names. All names with the same number of characters should be printed on the same line, and names with fewer characters should be printed first.
The runtime must be $O(n \log n)$ where n is the number of movies in the list.
You may use following map to store the string to be printed on the same line.
`Map<Integer, List<String>> map = new ...` // Choose the appropriate map class
Note: Use a single String to store more than one movie may slow down the runtime (but a `StringBuilder` is fine).
3. Write a function called `printDuplicates` that takes a List of names and print duplicate names from the list. The expected runtime must be $O(n)$. The name should be printed in the same order as it appears in the list. Don't call the list's `contains` method, because it is too slow.
4. Write a function called `sumZero` that takes an array of integers, then check whether there're two distinct number in the array to sum to zero, if so, return the index of first number. If not, return -1. The expected runtime must be $O(n)$ where n is the array length.

Extra Credit: Write a function called `longestZipZag` that returns how many edge in this longest zig-zag path.

A zig-zag path: 1. From node n , choose either left or right direction. 2. If the node is not null from this direction, there's one edge, and move to that child node. 3. From this child node, change direction and repeat all the steps. An empty tree's `longestZipZag` is 0.

Change the class name as `A3PrintName` follow by your initials at the end.

Change the [AVLtree](#) class name to include your initials at the end