FUNCTIONS	BEST-CASE	WORST-CASE
createZone	The best case would be when the tree is balanced. So, we just use the insert method from the BST so the complexity will be **O(log(n))**	The worst case is if the tree has a linear shape, so now it won't work as a tree but as a list, so the complexity will be **O(n)**
assignsDistributor	The best is when the zone tree is balanced, and the tree of dsmembers is balanced. Because we would have to find the dsmember and insert So, the complexity is **O(log(n))**	The worst case is when the zone tree is linear, and the tree of dsmembers is also linear. Because we would have to find the dsmember and insert. So, the complexity is **O(n)**
showDistibutor	In this case, there will be no best or worst case, because we must go through the entire dsmembers of the zone so, we end up with a complexity of **O(n)**	In this case, there will be no best or worst case, because we must go through the entire dsmembers of the zone so, we end up with a complexity of **O(n)**
deleteDistributor	The best case will be when the zone tree is balanced, the zone is a root(because finding it will just take a single step) and the dsmember is a leaf (removing it is also a single step), then we will have a complexity of **O(1)**	The worst case will be when the zone and dsmember tree has a linear structure, and we would have to go through all the elements to find the zone and dsmember to remove it so we have a complexity of **O(n)**
showZones	In this method there are no best or worst case as we would have to go through all the elements in the zone and dsmember tree using an inorder inside another inorder, so the complexity will be **O(n^2)**	In this method there are no best or worst case as we would have to go through all the elements in the zone and dsmember tree using an inorder inside another inorder, so the complexity will be **O(n^2)**
deleteZone	The best case is when the zone is not found, and the tree is balanced but we would have to go through all the elements in the tree so the complexity will be **O(log(n))**	The worst case will be when the number of zones is equal to the amount of dsmember we want to delete, and we would have to distribute all the dsmembers so we have a complexity of **O(n^2)**

isBalanced	The best case will be when the tree is not balanced from the root, and we must call the size method, so we have a complexity of **O(n)**	The worst case will be when the tree is balanced, so we would have to check each node and compute the size, so we would have a complexity of approximately **O(nlog(n))**
balance	The best case is when the tree is already balanced and we don't have to do any operation. The complexity will be **O(nlog(n))**	The worst case will be when the tree has a linear data structure, the complexity will be **O(n^2log(n))**