|  |
| --- |
| Employee |
| Employee ID |
| Company ID |
| Salary |
| Grade |
|  |
|  |

For later when we write DRY

|  |
| --- |
| DS |
| All time Highest Earner with smallest ID |
| AVL tree companies, in each company two AVL trees of it’s employees, by salary and by id |
| All Employees AVL tree by id  Used in RemoveEmployee  ,GetEmployeeInfo |
|  |
|  |
|  |

|  |
| --- |
| Company |
| Company id |
| Value |
| Max earner in company with smallest ID |
| Employees AVL tree by salary  Employees AVL tree by ID |
| Num of employees |
|  |

**AddCompany Function:**

We search the Companies AVL tree in log(k) to find the place for new company, insert company, and make necessary rotations.

**AddEmplyee Function:**

We search the Companies AVL tree in log(k) to find the Company then we go to the company’s employees AVL tree and we insert the new Employee. We also insert the Employee in the ALL employees AVL, perform necessary rotations in both trees, and perform checks like Highest earner, num of employees…

**RemoveEmployee Function:**

We search the All Employees AVL tree and find the specific employee ID, We delete both the employee object and the pointer to it , and perform necessary rotations in both AVL trees, and perform changes to highest earner, num of employees

**Remove Company Function:**

Only delete if companies num of employees in zero, Delete company from companies AVL tree and perform necessary rotations.

**GetCompanyInfo Function:**

Search Companies AVL tree for company, and return values

**GetEmployeeInfo Function:**

Search ALL Employees AVL tree for Employee, and return values.

**IncreaseCompanyValue Function:**

Search Companies AVL tree for Company, Change Value as instructed.

**PromoteEmployee** **Function:**

Search ALL employees for specific employee, Follow pointer, Change Salary and Grade as instructed.

**HireEmployee Function:**

We first search the all Employees fir the specified employee,

if they don’t exist, FAILURE

We search the Companies AVL for the specified Company

If it doesn’t exist, FAILURE

Else, temp copy Employee object, remove from both AVL trees, perform necessary rotations, perform necessary changes to num of employees, max earner.. then insert employee into new company

**OR**

Use RemoveEmployee, and AddEmployee functions

**AcquireCompany Function:**

We search for the two companies in the Companies AVL tree, and place two pointers to them.

we perform the value check to see if the transfer is valid,

If it is valid, merge two AVL trees using the algorithm discussed in: [Merge Two Balanced Binary Search Trees - GeeksforGeeks](https://www.geeksforgeeks.org/merge-two-balanced-binary-search-trees/)

Make necessary changes to value… etc

This requires our AVL tree to have inOrderTraversal functionality.

**GetHighestEarner**:

Use the AllTimeHighestEarnerSmallestID employee element in both ALL employees, or in the comapy’s employee tree

**GetAllEmployeesBySalary:**

**If company id is positive**

Use the companyEmployees AVL tree ordered by salary of asked company, traverse.

If not