Company Organizer using Trees, Heaps and Hashing Table

**EECS233**

**Group members:**

Justin Tang jzt5

Jielei Tiffany Li jxl1047

**Program Introduction:**

Our main concept here is reworking a binary heap to enter and store information about company employees, along with sorting algorithms that help the company to better organize its employee’s information.

**Program Structures:**

1. Each node in the heap/tree contains multiple attributes of information such as name of the employee, name of the company, the employee’s position, the employee’s experience and the salary.
2. A separate class called “Employee” in our program will help to organize and compare certain attributes among different employees and ultimately helps the main sorting algorithm in the “BinaryHeapX” Class.
3. Our algorithm is also designed to facilitate quick updates about any changes or taking new information/ attributes about the employee information. The corresponding method for updating new information is the “Percolating” algorithm. The employees are sorted from top down by their salary, with the smallest at the top and the largest at the bottom. The person with the highest salary would be recognized as the top employee at the company.
4. Our sorting algorithm also includes hashing the data from the heap to a array whereas the affiliated array functions as the primary database for information storation. Each employee is recognized by their name.
5. Ideally, we were going to have a hash table, but we found an array to be more feasible in this short period of time. Each employee would have had his or her name/ ID number as a key, with our Hashing algorithm, we could then be able to store the information in a hash table separate from the tree. Several other characteristics of the hash table also makes it the best choice for our Company Organizer althorim.
6. Our main method, “SortUI”, executes the sorting and then writes the data into .txt files for each specific attribute. It was also supposed to ask the user if they would like to add, delete, or edit any of the employees.
7. We have also enabled scanner method and txt data conversion method, which will be able to save separate txt files of information from the database.

**Sample Output of Sorting the employee’s information base on salary:**

**(Whereas you would also be able to sort the database base on any other attributes)**

**Saved data is seen on the .txt files**

Sort on salary:

Name: Ryan Howard

Company: Dunder Mifflin Paper Company

Position: Temp

Experience: 1

Salary: 20000

Name: Pam Beasley

Company: Dunder Mifflin Paper Company

Position: Receptionist

Experience: 4

Salary: 30000

Name: Kelly Kapoor

Company: Dunder Mifflin Paper Company

Position: Customer Service Rep

Experience: 3

Salary: 35000

Name: Jim Halpert

Company: Dunder Mifflin Paper Company

Position: Sales Rep

Experience: 4

Salary: 50000

Name: Toby Flenderson

Company: Dunder Mifflin Paper Company

Position: Human Resources Rep

Experience: 5

Salary: 55000

Name: Kevin Malone

Company: Dunder Mifflin Paper Company

Position: Accountant

Experience: 5

Salary: 60000

Name: Oscar Martinez

Company: Dunder Mifflin Paper Company

Position: Accountant

Experience: 6

Salary: 70000

Name: Dwight Schrute

Company: Dunder Mifflin Paper Company

Position: Assistant to the Regional Manager

Experience: 10

Salary: 70000

Name: Stanley Hudson

Company: Dunder Mifflin Paper Company

Position: Sales Rep

Experience: 9

Salary: 70000

Name: Phyllis Vance

Company: Dunder Mifflin Paper Company

Position: Sales Rep

Experience: 12

Salary: 75000

Name: Angela Martin

Company: Dunder Mifflin Paper Company

Position: Accountant

Experience: 8

Salary: 80000

Name: Creed Bratton

Company: Dunder Mifflin Paper Company

Position: Quality Assurance Director

Experience: 20

Salary: 100000

Name: Michael Scott

Company: Dunder Mifflin Paper Company

Position: Regional Manager

Experience: 15

Salary: 120000