CSE222 – Homework5 – Question4

Report

Nevzat Seferoglu

171044024

Problem Solution Approach :

* Problem Definition :

Solve the problem in Q3 by using a max_heap (where the maximum element is in the root node) this time. Implement your heap by using ArrayList as described in your book. Implement **MaxHeap** class to handle the ArrayList heap operations. Use the **AgeData** class you implemented in Q3 to hold age and the number of people at that age data. **The key of heap will be "number of people" this time. So, the age which the highest number of people is at, will be at the root.** Write a Comparator to compare two objects of class AgeData. You should use the compare method in your comparator to compare the elements in the heap. Implement the following methods (and any other methods you need) in MaxHeap class:

- **add** function to add a new record. It will first check if an AgeData object with that age exists in any index of the ArrayList. If it exists, the number of people field of the AgeData object in that node will be increased 1. (Check if any changes in heap is needed since the key is "number of people".) Otherwise a new heap record with the AgeData object will be inserted.
- While removing a node, the **remove** function will first check if a node with that age exists. If it exists and the number of people field of this node's AgeData object is greater than 1, it will decrease the number of people field 1. (Check if any changes needed since the key is "number of people".) If the number of people field is 1, it will remove the node.
- The **find** method will get an AgeData object of any age and find the AgeData object with the same age and return it.
- Add a **youngerThan** method which returns the number of people younger than an age.
- Add an **olderThan** method which returns the number of people older than an age.

* Approach :

Heap is an abstract data type. In this assignment object of AgeData has been handled in the heap. Heap can be manipulated with index arithmetic. In the nodes are held in certain index and each node except root has parent node. We can say that heap has a same effect with binary tree. This property provides an easy way for handling data with indexes.

Heap Right Child = 2(index) + 1

Heap Right Child = 2(index) + 2

Parent = (x – 1) / 2

Test Cases:

Constructors Test:

Test ID	Scenario	Test Data	Expected Result	Actual Result	Pass	Fail
T01	Create heap with constructor.	Created heap object.	Create heap successfully.	Expected result has occurred.	✓	

Methods Test:

Test ID	Scenario	Test Data	Expected Result	Actual Result	Pass	Fail
T01	- add(new AgeData(10)) - add(new AgeData(20)) - add(new AgeData(5)) - add(new AgeData(15)) - add(new AgeData(10)) - add(new AgeData(9)) - add(new AgeData(25)) - add(new AgeData(16)) - add(new AgeData(18)) - add(new AgeData(20)) - add(new AgeData(9)) - add(new AgeData(9))	Each of them is representing AgeData reference. 10 , 20 , 5, 15, 10,9,25,16,18	Construct a heap as a binary heap.	Expected result has occurred.	✓	
T02	find(new AgeData(10)).toString()	10 , 20 , 5, 15, 10,9,25,16,18	10 - 2	Expected result has occurred.	✓	
T03	- find(new AgeData(18)).toString()	10 , 20 , 5, 15, 10,9,25,16,18	18 - 1	Expected result has occurred.	✓	
T04	- find(new AgeData(7)).toString()	10 , 20 , 5, 15, 10,9,25,16,18	Given item has not been found in heap.	Expected result has occurred.	✓	
T05	- youngerThan(15)	10 , 20 , 5, 15, 10,9,25,16,18	6	Expected result has occurred.	✓	
T06	- olderThan(10)	10 , 20 , 5, 15, 10,9,25,16,18	6	Expected result has occurred.	✓	
T07	- remove(new AgeData(9))	10 , 20 , 5, 15, 10,9,25,16,18	Remove the AgeData object from the heap.	Expected result has occurred.	✓	
T08	- remove(new AgeData(10))	10 , 20 , 5, 15, 10,9,25,16,18	Remove the AgeData object from the heap.	Expected result has occurred.	✓	
T09	- remove(new AgeData(7))	10 , 20 , 5, 15, 10,9,25,16,18	Item cannot be found and removed.	Expected result has occurred.	✓	

Class Diagram :

Class diagram is inserted to the homework file.

Running Commend and Result:

Running command and result stage is inserted to the homework file.