**GTÜ Department of Computer Engineering**

**CSE 222/505 - Spring 2021**

**Homework 4 Report**

**FATİH DOĞAÇ**

**1901042654**

**1.System Requirements**

User must have Java Runtime Environment to run the program.

It can be found [here.](https://www.java.com/en/download/manual.jsp)

**Functional Requirements:**

User can build a Max-Heap with following features:

* Can add any comparable type of value to the heap.
* Can remove a value from the heap.
* Can remove the i’th largest wanted element from the heap.
* Can merge with another heap.
* Can search the heap for a specific item.
* Can set a data that returned from the iterator.
* Can print the heap.
* Can know the number of elements in the heap.

User can build a BSTHeapTree with following features:

* Can add any comparable type of value to the tree.
* Can remove a value from the tree.
* Can search for a value in the tree.
* Can find the mode of the tree.
* Can print the tree.
* Can know the number of elements in the tree.

There is a driver program for user in the file to see features clearly.

**Text

Description automatically generated2.Class Diagram:**

**3.Problem Solution Approach**

I had several problems during the creating the project progress.

1. After deleting an element, if the heap that element removed from will become empty , That heap (in other words that binary tree node) needs to deleted. And my method was giving segmentation fault.
2. Keeping the occurrences of the values.

For the first problem, I needed to go deep in the structure and implemented myself a simple binary search tree to understand the node delete method.

Then I fully understood the algorithm then implemented for BSTHeapTree.

For the second problem , after a few hours of thinking and trying to make up something, I came up with an idea of creating an another class which is ValueOccurance. ValueOccurance holds a data and its occurrence. That solved my problem I created the heaps with ValueOccurances in them.

**4.TEST CASES**

1. Insert the 3000 numbers that are randomly generated in the range 0-5000 into the

BSTHeapTree. Store these numbers in an array as well. Sort the numbers to find the

number occurrences of all the numbers.

2. Search for 100 numbers in the array and 10 numbers not in the array and make sure

that the number of occurrences is correct.

3. Find the mode of the BSTHeapTree. Check whether the mode value is correct.

4. Remove 100 numbers in the array and 10 numbers not in the array and make sure that the number of occurrences after removal is correct.

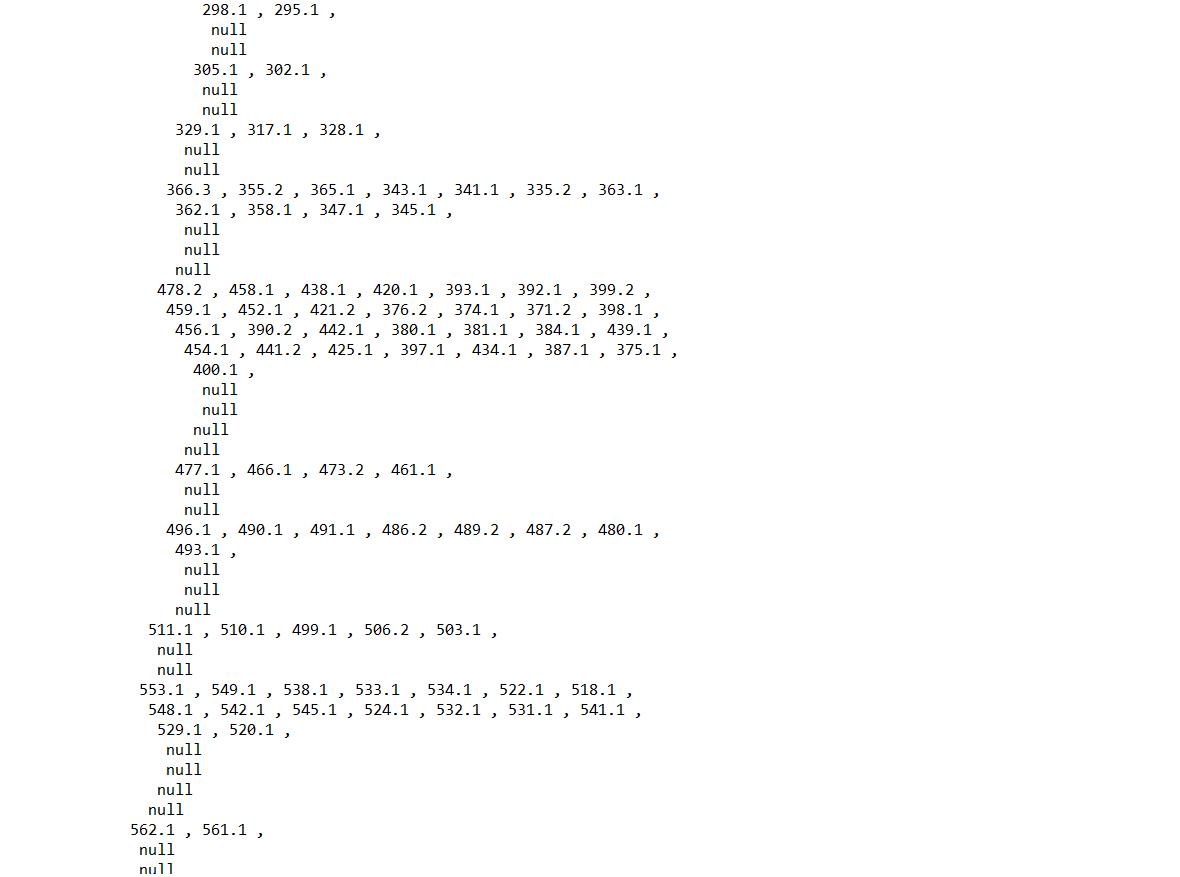
**5. RUNNING AND RESULTS**

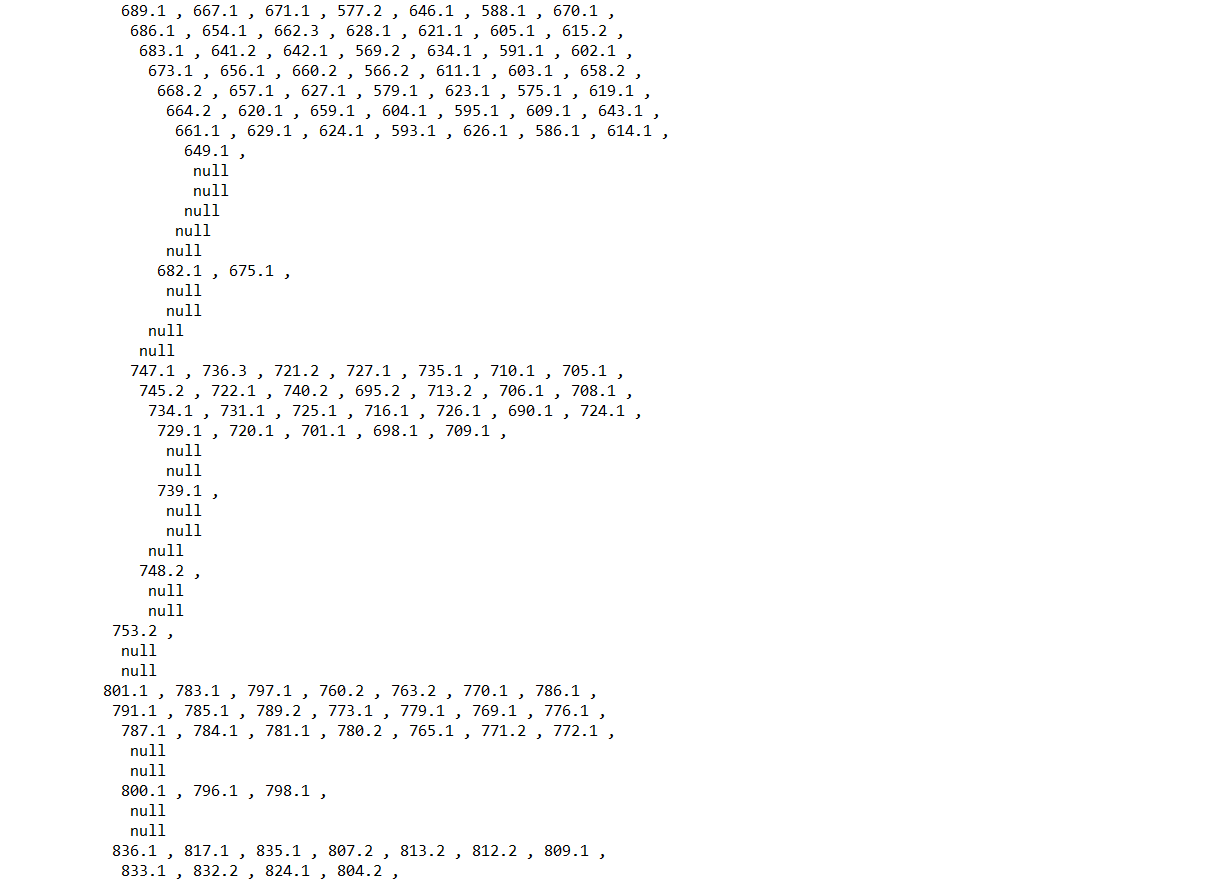
**Graphical user interface, text, application

Description automatically generated**

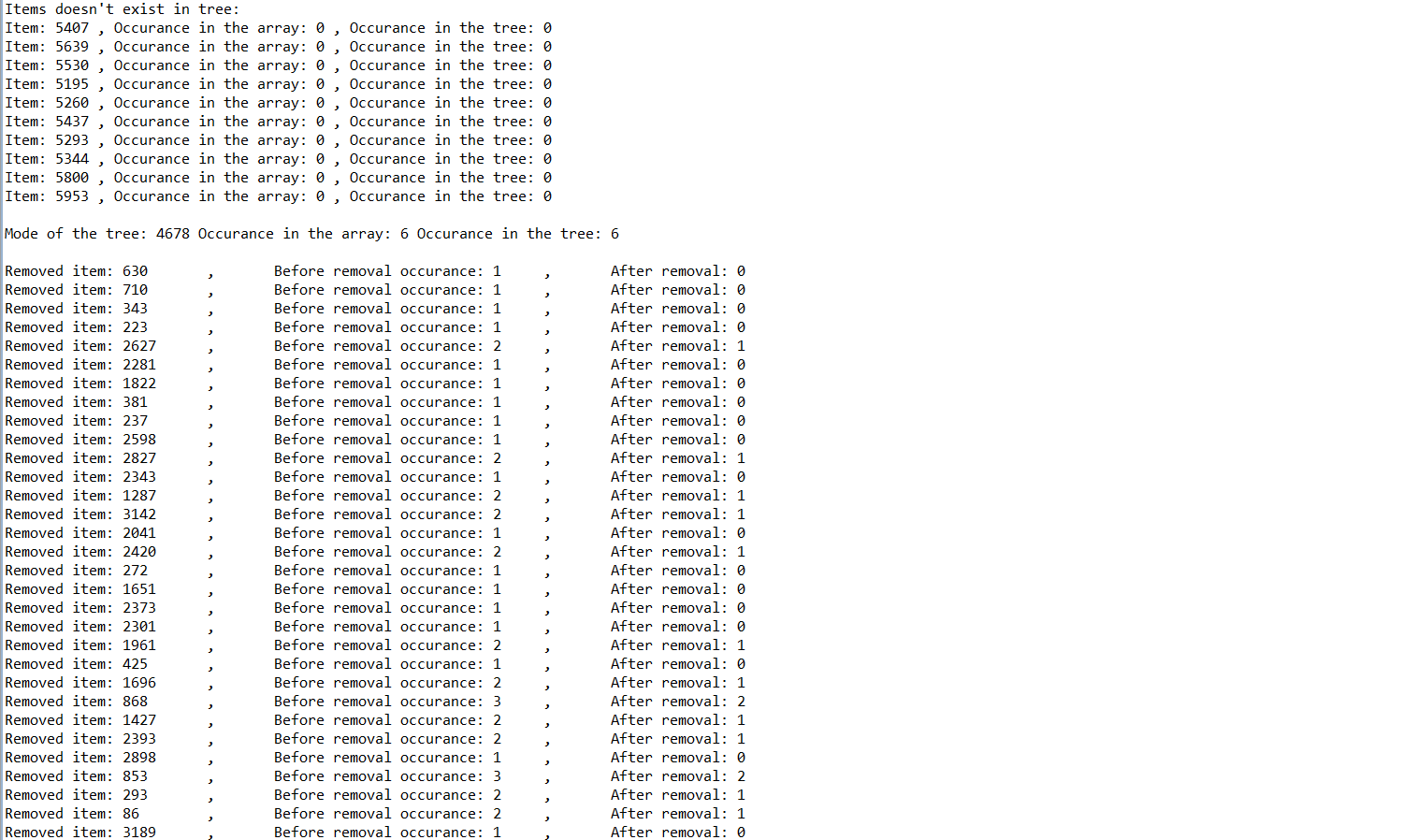
**Scatter chart

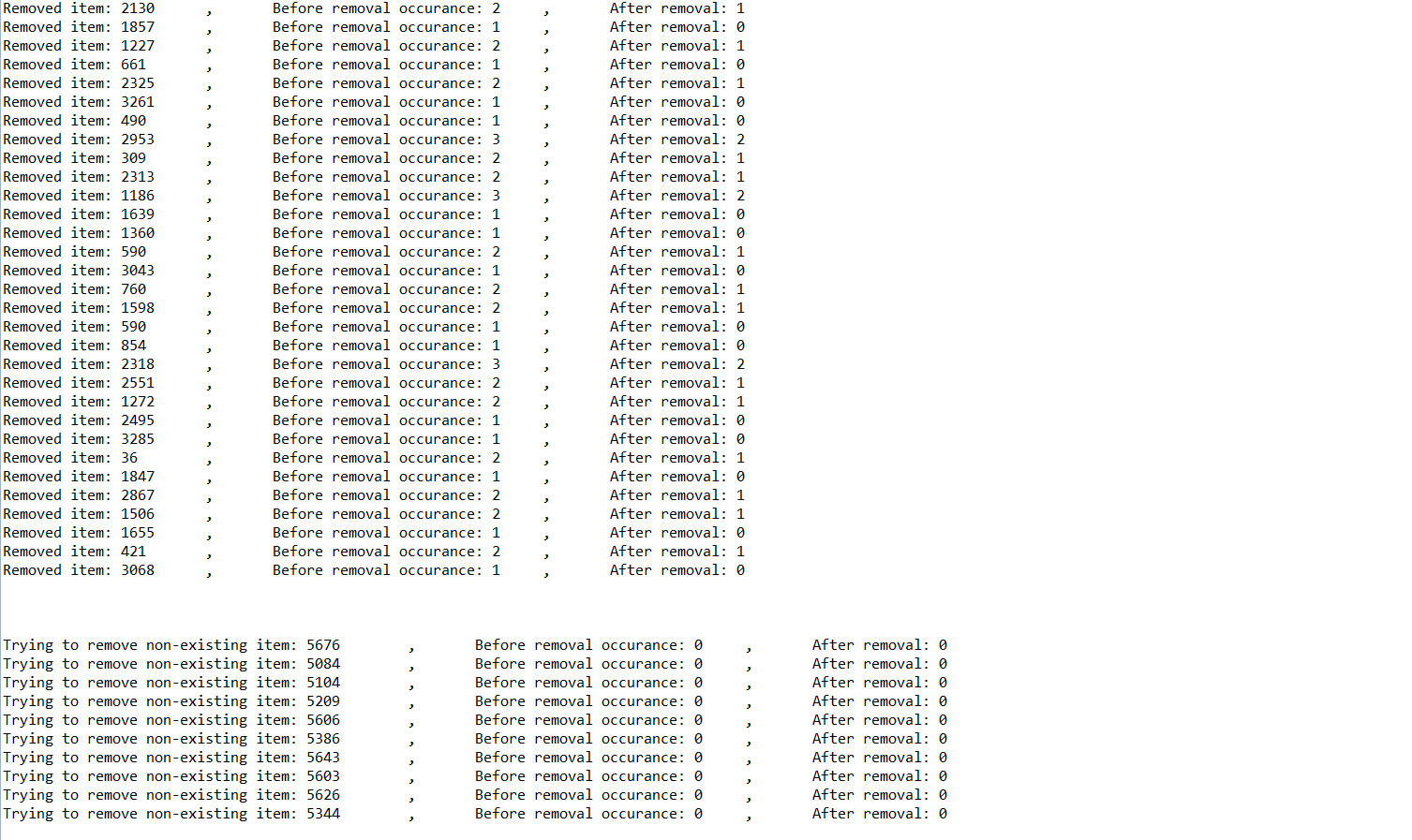
Description automatically generated**

****

****

****

****

****