T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY of ENGINEERING
DEPARTMENT of COMPUTER ENGINEERING

Heap and BSTHeapTree

CSE 222 DATA STRUCTURES AND ALGORITHMS
HOMEWORK 4 REPORT

STUDENT
Şeyda Nur DEMİR
12 10 44 042

LECTURER
Prof. Dr. Fatih Erdoğan SEVİLGEN

TEACHING ASSISTANT
Başak KARAKAŞ

KOCAELİ, 2021

1.DESCRIPTION

This homework has three parts.

PART 1: 40 pts.

You should implement the following features for the Heap structure.
i. Search for an element
ii. Merge with another heap
iii. Removing ith largest element from the Heap
iv. Extend the Iterator class by adding a method to set the value
(value passed as parameter) of the last element returned by the next methods.

RESTRICTIONS

- Can be only one main class in project
- Don't use any other third part library

General Rules

- For any question firstly use course news forum in Moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over sixty percent (%60).

Technical Rules

- You must write a driver function that demonstrates all possible actions in your homework. For example, if you are asked to implement an array list and perform an iterative search on the list then, you must at least provide the following in the driver function :
  - Create an array list and add items to the list. Append items to head, tail, and kth index of the list.
  - Perform at least two different searches by using two items in the list and print the index of the items.
  - Perform another search with an item that isn't in the array list and inform the user that the item doesn't exist in the array list.
  - Delete an existing item from the list and repeat the searches.
  - Try to delete an item that is not on the array list and throw an exception for this situation.

  The driver function should run when the code file is executed.
- Implement clean code standards in your code ;
  - Classes, methods and variables names must be meaningful and related with the functionality.
  - Your functions and classes must be simple, general, reusable and focus on one topic.
  - Use standard java code name conventions.

Report Rules

- Add all javadoc documentations for classes, methods, variables …etc. All explanation must be meaningful and understandable.
- You should submit your homework code, Javadoc and report to Moodle in a "studentid_hw4.tar.gz" file.
- Use the given homework format including selected parts from the table below:
  - Detailed system requirements
  - The project use case diagrams (extra points)
  - Class diagrams
  - Other diagrams (no need for this assignment)
  - Problem solutions approach
  - Test cases
  - Running command and results

Grading

- No OOP design: -100
- No interface: -95
- No method overriding: -95
- No error handling: -50
- No inheritance: -95
- No polymorphism: -95
- No javadoc documentation: -50
- No report: -90
- Disobey restrictions: -100
- Cheating: -200
- Your solution is evaluated over 100 as your performance.

2.REPORT

I detailed here what I did in my project.
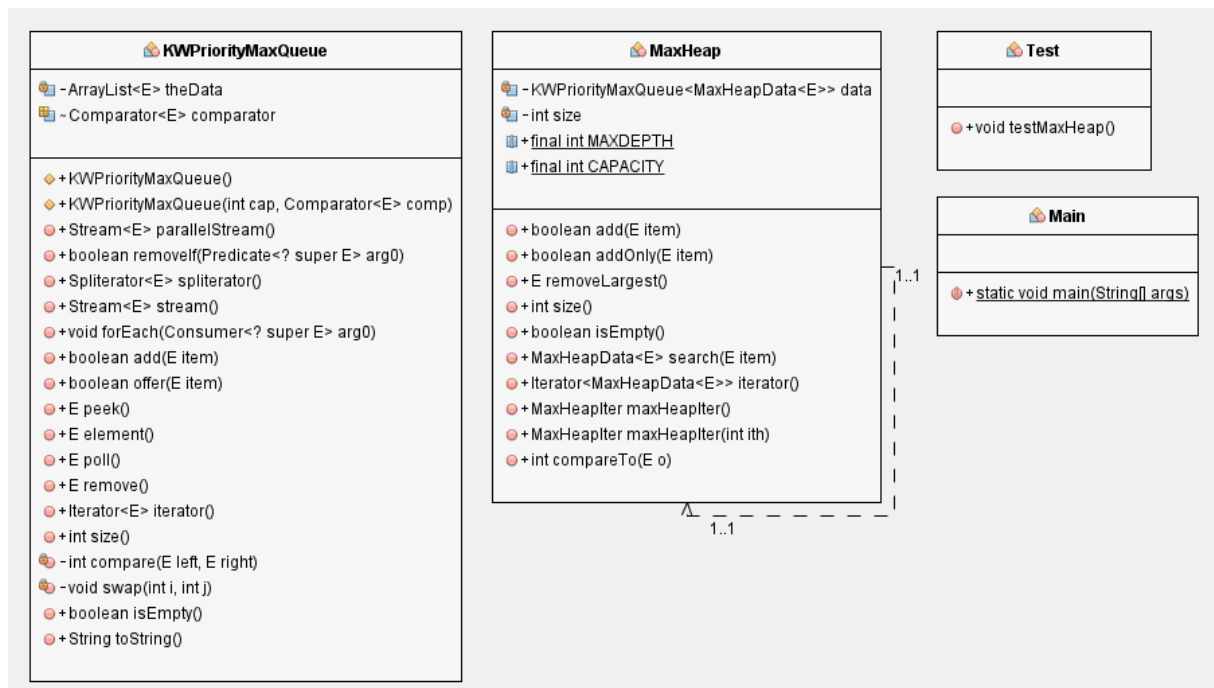
2.1.System Requirements

Requirements

Heap Operations :
- Search
- Merge
- Remove Largest
- Iterator Set

2.2.Use Case Diagrams

I did not draw use case diagram.

2.3.Class Diagrams

KWPriorityMaxQueue

- ArrayList<E> theData
- ~Comparator<E> comparator

- +KWPriorityMaxQueue()
- +KWPriorityMaxQueue(int cap, Comparator<E> comp)
- +Stream<E> parallelStream()
- +boolean removeIf(Predicate<? super E> arg0)
- +Spliterator<E> spliterator()
- +Stream<E> stream()
- +void forEach(Consumer<? super E> arg0)
- +boolean add(E item)
- +boolean offer(E item)
- +E peek()
- +E element()
- +E poll()
- +E remove()
- +Iterator<E> iterator()
- +int size()
- -int compare(E left, E right)
- -void swap(int i, int j)
- +boolean isEmpty()
- +String toString()

MaxHeap

- KWPriorityMaxQueue<MaxHeapData<E>> data
- int size
- final int MAXDEPTH
- final int CAPACITY

- +boolean add(E item)
- +boolean addOnly(E item)
- +E removeLargest()
- +int size()
- +boolean isEmpty()
- +MaxHeapData<E> search(E item)
- +Iterator<MaxHeapData<E>> iterator()
- +MaxHeapIter maxHeapIter()
- +MaxHeapIter maxHeapIter(int ith)
- +int compareTo(E o)

Test

- +void testMaxHeap()

Main

- +static void main(String[] args)

1..1

1..1

8

2.4.Other Diagrams

I did not draw any other diagram.

2.5.Problem Solutions Approach

Note : I googled the way you said in the report to problem solving approach, but I could not any useful article, what I found was either paid or long. After all I found a useful post from medium. I prepared this part according to this post. I hope I got it right.

- **Clearly understanding and/or defining the problem :**
  I understood and defined the problem clearly.
- **Breaking down large problems into smaller problems :**
  I broke down large problem into smaller problems.
- **Solving the problem at an abstract level first :**
  I solved the problem at an abstract level first.
  - o I thought a lot about the problem.
  - o I scribbled something about this subject in the ledger.
  - o Something started to take shape in my head.
  - o I used my knowledge of data structures, and it is.
- **Using notes and pseudo-code :**
  I noted what I thought, and wrote permanently pseudo-codes mixed with java codes.
  - o If i understand or find something new, I noted.
  - o I wrote pseudo-codes mixed with java codes, not clear.
- **Running code early and often :**
  I wrote real codes and run them often.
  - o I turned my pseudo-codes into real java codes.
  - o I coded them in ide and run them often.

2.6.Test Cases

Testing Requirements

Heap Operations :

- Search (done)
- Merge
- Remove Largest (done)
- Iterator Set (done)

2.7.Running Commands and Results

Compile and Run Commands, Testing Steps

Usage of my heap program :

1. Compile program with "**javac *.java**" command
2. Run program with "**java Main**" command
3. Program runs the driver code, then exits.

PART 3 : TIME COMPLEXITY ANALYSIS

Heap Structure

END OF THE REPORT

LAST UPDATE

May 05, 2021   Friday   23:50

STUDENT
Şeyda Nur DEMİR
12 10 44 042

LECTURER
Prof. Dr. Fatih Erdoğan SEVİLGEN

TEACHING ASSISTANT
Başak KARAKAŞ

KOCAELİ, 2021