CSCI 321 Computer Science III Summer 2019

Assignment 3

***Maintain a task heap***

You have a set of tasks and each task has its priority. A task is a pair [String name, Integer priority]. For example, [Task1, 5], [Task2, 2]. We need to add these tasks into a priority queue. As we know, heaps are usually used as the underlying data structures of priority queues. In Java, there are several constructors for implementing a priority queue. Please see the reference below:

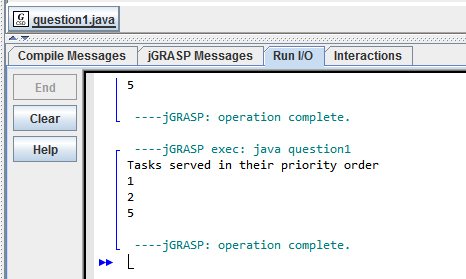
https://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html

Part a. Regarding the default priority queue constructor, can you show it (PriorityQueue) is implemented with Min-Heap, that is the top element is the minimum one in the heap. Write a program to demonstrate. Attach your code and screenshots of the output. You need to add a set of tasks (with priority) into the priority queue first. For example, you add [Task1, 5], [Task2, 2], [Task3, 1], and you should show (print out) the following in your output after utilizing the “poll” method (check the reference link above for details).

[Task3, 1]

[Task2, 2]

[Task1, 5]



import java.util.\*;  
  
class question1

{  
 public static void main(String[] args)  
 {  
 Scanner in = new Scanner(System.in);   
 PriorityQueue<Integer> test =new PriorityQueue<Integer>(5) ;  
 Task Task1 = new Task("Task1", 5);  
 test.add(Task1.priority);  
 Task Task2 = new Task("Task2", 2);  
 test.add(Task2.priority);   
 Task Task3 = new Task("Task3", 1);  
 test.add(Task3.priority);  
   
 System.out.println("Tasks served in their priority order");  
   
 while (!test.isEmpty()) {  
 System.out.println(test.poll());  
 }  
 }  
}  
  
  
class Task

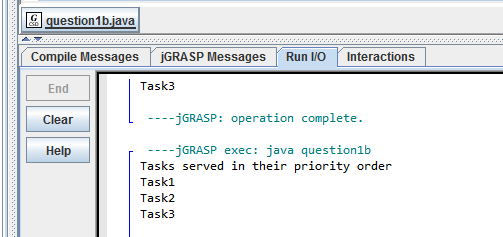
{  
 public String name;  
 public int priority;  
   
 public Task(String name, int priority)   
 {   
 this.name = name;  
 this.priority = priority;  
 }  
   
 public String getName()   
 {  
 return name;  
 }  
}

Part b. Can you construct a Max-Heap using a comparator? Please review the sample code in Lecture 4 Activity 1. Write a program to demonstrate. Attach your code and screenshots of the output. For example, you add [Task1, 5], [Task2, 2], [Task3, 1], and you should show (print out) the following in your output after utilizing the “poll” method (check the reference link above for details).

[Task1, 5]

[Task2, 2]

[Task3, 1]



import java.util.\*;  
  
class question1b   
{  
 public static void main(String[] args)  
 {  
 Scanner in = new Scanner(System.in);  
 PriorityQueue<Task> test = new  
 PriorityQueue<Task>(5, new TaskComparator());  
 Task Task1 = new Task("Task1", 5);  
 test.add(Task1);  
 Task Task2 = new Task("Task2", 2);  
 test.add(Task2);   
 Task Task3 = new Task("Task3", 1);  
 test.add(Task3);  
   
 System.out.println("Tasks served in their priority order");  
   
 while (!test.isEmpty()) {  
 System.out.println(test.poll().getName());  
 }  
 }  
}  
  
class TaskComparator implements Comparator<Task>  
{  
 public int compare(Task s1, Task s2)   
 {  
 if (s1.priority < s2.priority)  
 return 1;  
 else if (s1.priority > s2.priority)  
 return -1;  
 return 0;  
 }  
 }  
  
class Task   
{  
 public String name;  
 public int priority;  
   
 public Task(String name, int priority)   
 {  
 this.name = name;  
 this.priority = priority;  
 }  
   
 public String getName()   
 {  
 return name;  
 }  
}