CSCI 321 Computer Science III Fall 2018

Assignment 3

Problem: ***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]

\*\*\*Please note….I set up a priority queue list and used that list to drive desired inputs from an array prebuilt with task no and values.

import java.util.PriorityQueue;

import java.util.\*;

public class partA {

public static void main(String[] args) {

int[][] taskValue = {{1, 5}, {2, 2}, {3, 1}};

PriorityQueue<Integer> taskPr = new PriorityQueue<>(10, Collections.reverseOrder());

taskPr.add(1);

taskPr.add(2);

taskPr.add(3);

System.out.println("Priority que after using poll():");

while (taskPr.size() != 0) {

int val = taskPr.poll() -1;

System.out.println("task" + taskValue[val][0]+ "," + taskValue[val][1]);

}

}

}

A screenshot of a computer

Description generated with very high confidence

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.Comparator;

import java.util.PriorityQueue;

public class partB {

public static void main(String[] args) {

Comparator<String> comparator = new StringLengthComparator();

PriorityQueue<String> queue = new PriorityQueue<String>(10, comparator);

queue.add("task1, 5");

queue.add("task2, 2");

queue.add("task3, 1");

while (queue.size() != 0) {

System.out.println(queue.poll());

}

}

}

// StringLengthComparator.java

import java.util.Comparator;

public class StringLengthComparator implements Comparator<String> {

@Override

public int compare(String x, String y) {

// Assume neither string is null. Real code should

// probably be more robust

// You could also just return x.length() - y.length(),

// which would be more efficient.

if (x.length() < x.length() + 1) {

return 1;

}

if (x.length() > x.length() + 1){

return -1;

}

return 0;

}

}

A screenshot of a cell phone

Description generated with very high confidence