

WIA1002/WIB1002 Data Structure

Tutorial: Priority Queue

1. Describe the main difference between Queue and PriorityQueue.
 - A queue follows **First-In, First-out (FIFO)** ordering where an item is removed from the queue is the first element that was added into the queue. (First come, first served)
 - A priority queue follows **Largest-in, First-out** behavior where the elements are assigned with priorities and the element with the highest priority is removed first.

2. Briefly provide THREE (3) real-life example in using PriorityQueue.
 - i. Hospital emergency rooms, the patients with life-threatening conditions get the highest priority to be treated compared to patients with less severity of their illness or injuries regardless of their arrival time.
 - ii. To-do list (*Task scheduling*), the tasks with the closest deadlines are at the front of the queue, ensuring they are handled first.
 - iii. Covid-19 Pandemic Vaccination Prioritization, the frontliners have a higher priority in taking offered vaccination slots before the public based on their role and risk level.

1. Describe the main difference between Queue and PriorityQueue.

A regular queue is a first-in and first-out (FIFO) data structure. Elements are appended to the end of the queue and are removed from the beginning of the queue. In a priority queue, elements are assigned with priorities. When accessing elements, the element with the highest priority is removed first.

2. Briefly provide THREE (3) real-life example in using PriorityQueue.

- a. Printing priority based on position of the sender in a department
 - to handle the jobs sent to the Computer Science Department's printer: Jobs sent by the department chair should be printed first, then jobs sent by professors, then those sent by graduate students, and finally those sent by undergraduates. The values put into the priority queue would be the priority of the sender (e.g., using 4 for the chair, 3 for professors, 2 for grad students, and 1 for undergrads), and the associated information would be the document to print. Each time the printer is free, the job with the highest priority would be removed from the print queue, and printed.
- b. Emergency department in Hospital – severe patient will be treated first.
- c. Airport check-in services – Gold class member will be served first.

3. Show the output for every `System.out.println` ((a) – (f)) in the following code:

```
import java.util.*;

public static void main(String args[]) {

    PriorityQueue<String> pQueue = new PriorityQueue<String>();

    pQueue.offer("C++");
    pQueue.offer("Python");
    pQueue.offer("Java");
    pQueue.offer("Fortran");

    System.out.println("peek() gives us: " + pQueue.peek()); // (a)
    System.out.println("The queue elements:"); // (b)

    Iterator itr = pQueue.iterator();
    while (itr.hasNext())
        System.out.println(itr.next()); // (b)

    pQueue.poll();
    System.out.println("After poll():"); // (c)
    Iterator<String> itr2 = pQueue.iterator();
    while (itr2.hasNext())
        System.out.println(itr2.next()); // (c)

    pQueue.remove("Java");
    System.out.println("After remove():"); // (d)
    Iterator<String> itr3 = pQueue.iterator();
    while (itr3.hasNext())
        System.out.println(itr3.next()); // (d)

    boolean b = pQueue.contains("Ruby");
    System.out.println("Priority queue contains Ruby or not?: " + b); // (e)

    Object[] arr = pQueue.toArray();
    System.out.println("Value in array: "); // (f)

    for (int i = 0; i < arr.length; i++)
        System.out.println("Value: " + arr[i].toString()); // (f)
}
```

(a)

peek() gives us: C++

(b)

The queue elements:

C++

Fortran

Java

Python

(c)

After poll():

Fortran

Python

Java

(d)

After remove():

Fortran

Python

(e)

Priority queue contains Ruby or not?: false

(f)

Value in array:

Value: Fortran

Value: Python

- a. peek() gives us: C++
- b. The queue elements:
C++
Fortran
Java
Python
- c. After poll():
Fortran
Python
Java
- d. After remove():
Fortran
Python
- e. Priority queue contains Ruby or not?: false
- f. Value in array:
Value: Fortran
Value: Python

4. Answer the following sub-questions with referring to the following code:

```
public class PriorityQueue2 {
    public static void main(String... args ){
        PriorityQueueComparator pqc=new PriorityQueueComparator();
        PriorityQueue<String> pq=new PriorityQueue<String>(5,pqc);
        pq.add("Jason");
        pq.add("Ali");
        pq.add("Muhamad");
        for(String s:pq){
            System.out.println(s);
        }
    }
}

public class PriorityQueueComparator implements Comparator<String>{
    public int compare(String s1, String s2) {
        if (s1.length() < s2.length()) {
            return -1;
        }
        if (s1.length() > s2.length()) {
            return 1;
        }
        return 0;
    }
}
```

- a) What is the purpose of the PriorityQueueComparator in the code?

The purpose of the PriorityQueueComparator is to define a custom ordering method for the elements of the PriorityQueue by overriding its default natural ordering. In this case, this custom comparator enforces length-based ordering in the PriorityQueue where the shortest string length has the highest priority to be the first.

- b) What is the output for the code?

Ali
Jason
Muhammad

- a) What is the purpose of the PriorityQueueComparator in the code?

Compare the length of the newly added element, return false if s1 is shorter than s2, return true if s1 is longer than s2.

- b) What is the output for the code?

Ali
Jason
Muhamad