

Tafita Rakotozandry  
CS150 Lab  
10/17/2020

## **Lab 9 Report**

### **INTRODUCTION**

In this lab, we created a priority queue based on an ArrayList. We used that new implemented priority queue to compare with the Java built in priority queue. The goal is to observe from an experiment how a heap data structure is very efficient for priority queue. The object that was inserted in the queue belongs to the class WeightElement. It was another class created in this lab.

### **Unit Tests**

Each of the priority queue methods were unit tested and compared with the Java's built in priority queue. Each test is made in a way that it tests a large queue and an almost empty queue to verify that every method behaves as they are intended to be.

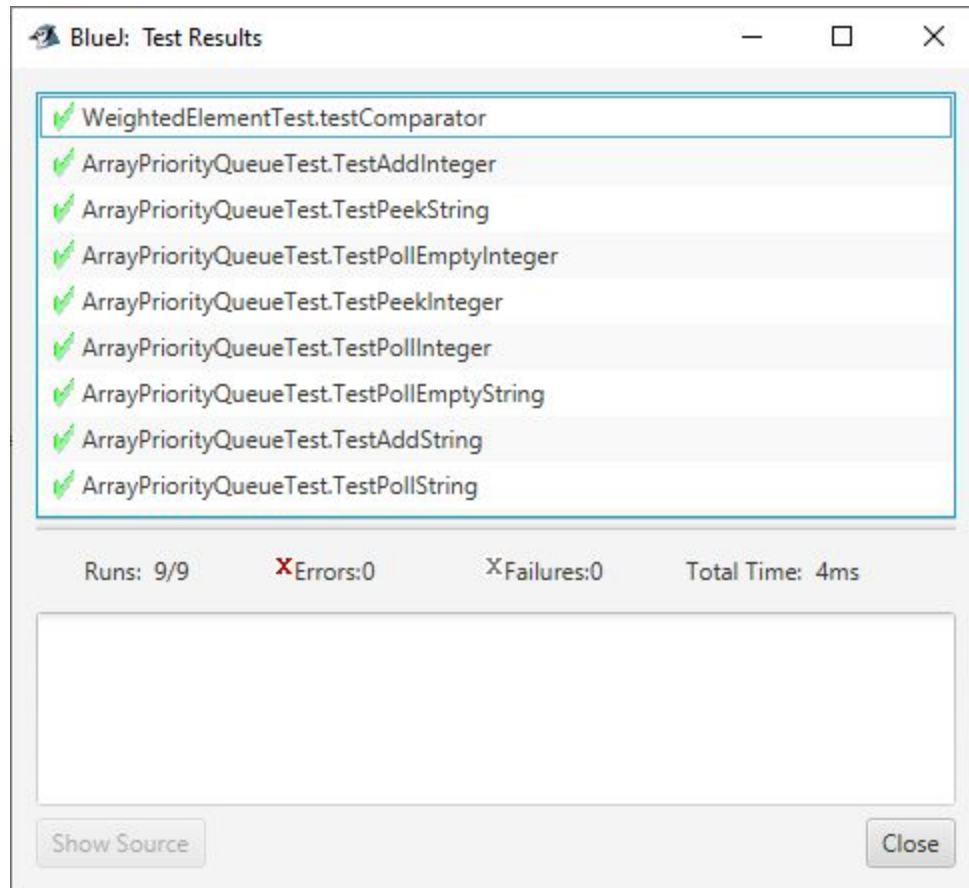


Figure 1: Screenshot Unit Testing Window

## Required Output

As part of the lab assignment , we were asked to compare the time to create the regular priority queue and our version of priority queue. The time to peek and to poll for each cases were also analyzed. Here are the results.

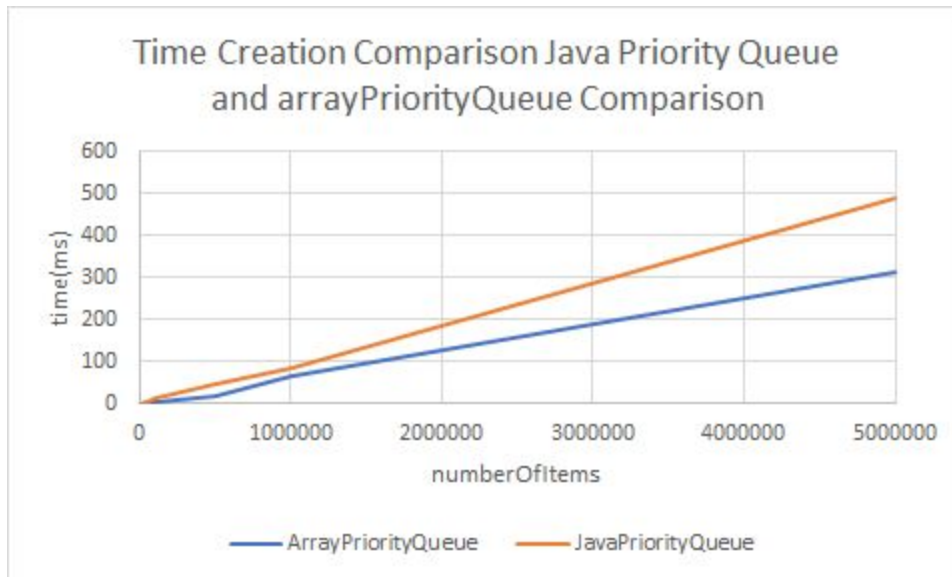


Figure 2: Output Graph

It takes much more time to create the default priority queue than our version of priority queue. However, the performance is so different when it comes to peek and to poll as presented below:

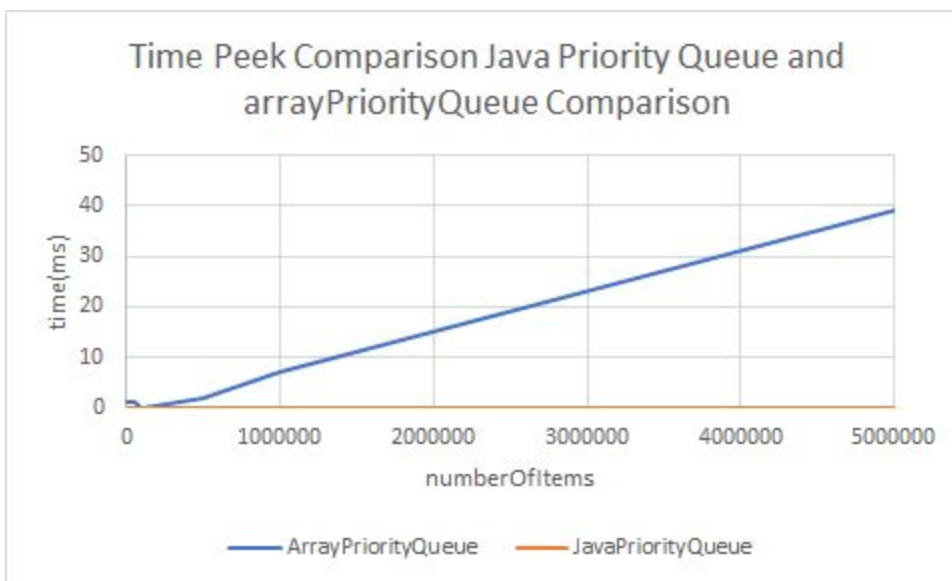


Figure 3: Output Graph

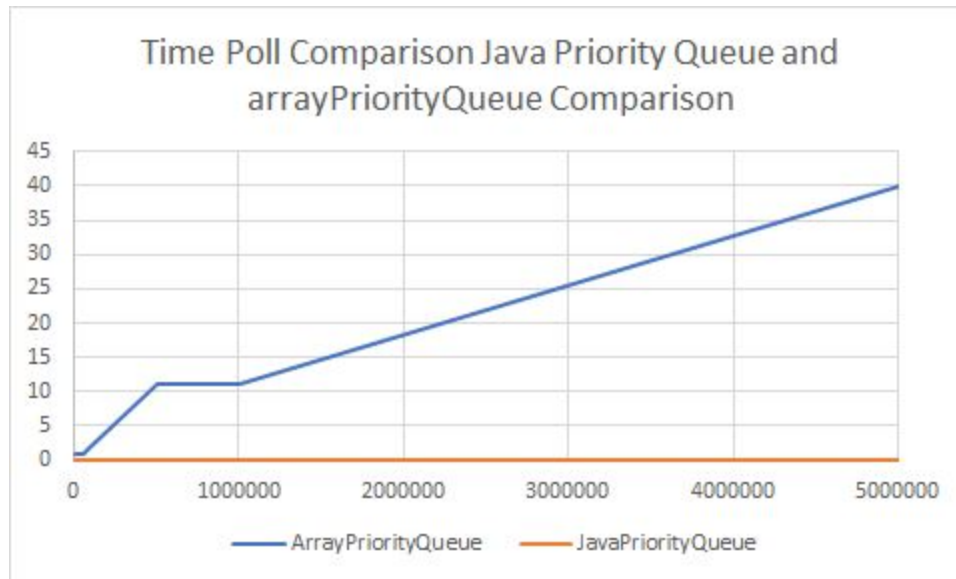


Figure 4: Output Graph

To perform a peek or a poll for the default priority queue is way more efficient than our version, which heavily depends on the data size. Hence, we can conclude that using a heap data structure is better than a simple arrayList.

### Trouble Report

I did not encounter any major issue in this lab. I just added a feature to write the results on a csv file in my experiment controller. I think it is a more efficient way to graph the output rapidly.

## References

*Michael Kölling (2015). Unit Testing in BlueJ <https://www.bluej.org/tutorial/testing-tutorial.pdf>*

*Weiss, M. A. (1998). Data structures and problem solving using Java. ACM SIGACT News, 29(2), 42-49.*