Data Structures & Algorithms:

John Vos

2018

# Index:

Contents

[Index: 1](#_Toc529722004)

[Proposed Data Structure: 2](#_Toc529722005)

[References: 3](#_Toc529722006)

## Proposed Data Structure:

Depending on the operations you intend to carry out you need to choose an implementation that is suitable. Therefore, for this assignment I decided to use a priority queue which I then implemented as a singly linked list as my data structure. I will now explain why:

First, when it comes to iterating over either kind of data type is practically equally cheap. The main benefit I found with using a linked list is the fact that I needed to use iterators to insert, remove and search elements in a sequential order and for this reason a linked list is the best option; as the list is dynamically resized as it is changed and manipulated, moreover, with insertion being of O(1) in an array the list can only be changed locally & needs to be moved (i.e. copied). What’s more, as I am searching through a linked list and the list is ordered in sequential order going from one link to the next link it will do it in O(n), therefore being more time efficient and appropriate than an array.

Secondly, if I had used an array, I would have had to declare the size of the array in the beginning, which as a result, would have allocated a set amount of memory for the array. Therefore, taken up and wasted memory and empty space as I did not know how many tickets I would need to create and put into the array at any given time.

Additionally, a linked list offers far more flexibility than a queue, this is due to the fact that to insert tickets into the correct order of priority I would need to remove every single element that is behind the current ticket or in the front to achieve the desired order of priority; thus in terms of time complexity, and moreover, efficiency this would result in a significant increase of time and memory consumption as more elements are being inserted.

# References:

(JavaPoint), S. I. P. L., 2011-2018. *Java Point.* [Online]   
Available at: https://www.javatpoint.com/collections-in-java  
[Accessed 11 2018].

Anon., 2017. *Geek for Geeks.* [Online]   
Available at: https://www.geeksforgeeks.org/linked-list-set-1-introduction/

Anon., n.d. *Geeks for Geeks.* [Online]   
Available at: https://www.geeksforgeeks.org/linked-list-set-2-inserting-a-node/?fbclid=IwAR39VBOH7Ch8-pSYNx7JmTjHUYbaGp\_xkKjPIWIJ0GUcZ1eu394PWWSPvdc

Chambers, S., n.d. *Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/322715/when-to-use-linkedlist-over-arraylist-in-java

Gabry, O. E., 2016. *Medium.* [Online]   
Available at: https://medium.com/omarelgabrys-blog/data-structures-a-quick-comparison-6689d725b3b0

Khandelqal, A., 2017. *geeks for geeks (practice).* [Online]   
Available at: https://practice.geeksforgeeks.org/problems/advantages-and-disadvantages-of-hash-table

Singh, Y., 2015. *My Learning.* [Online]   
Available at: http://www.mylearning.in/2015/06/pros-and-cons-of-different-data.html  
[Accessed 11 2018].