Priority Queues and Heaps

Sujit Chakrabarti

1 Real Life Example – Selecting Privileged Bank Customers

In previous videos we have learned about FIFO queues and have observed that there are many examples in real-life where they occur: movie ticket queue, to-do lists etc.

However, we know from our practical experience, we know that often these queues don't follow the FIFO order. For example, in a to-do list, a task introduced later in the list may have to be completed earlier based upon its urgency or importance. Similarly, if it's a checkin counter queue in an airport where there are passengers of a flight which is about to depart, these passengers may have to be allowed to jump the queue. In other words, the real life queues are often not FIFO queues but priority queues. In these queues, which element will the next one to be dequeued isn't necessarily decided based on the time at which it was enqueued, but on some other criteria as well.

1.1 Java Code

(code: `pq1`, `pq5`)

We represent the accounts with integers denoting their balance amount.

2 Priority Queue ADT

1. *removeMinimum*
2. *minimum*
3. *add*
4. *isEmpty*
5. *size*

3 PQ using Lists

3.1 Fast add, slow remove

(selection sort)

- Add in the end ($O(1)$)
- Search and remove the minimum ($O(n)$)

3.2 Slow add, fast remove

(insertion sort)

- Search and insert to maintain sorted order ($O(n)$). In array list, binary search will reduce the time to search to $O(log(n))$ but will increase the actual insertion step to $O(n)$ leading to the same performance.
- Remove the minimum from the head ($O(1)$)

3.3 Effect of Mutable fields

The functional characteristics of the priority queue may vary from implementation to implementation if the fields being compared are mutable.

3.4 Generic PQs

1. Using `java.lang.Comparable` (code: `pq2`)
2. Using `java.util.Comparator` (code: `pq3`)

3.5 Performance Analysis

4 Heap

(code: `pq4`)