

Assignment 3 (mandatory)

Airport Queueing

Jacob Trier Frederiksen (instructor)
& Anders Kalhauge (AKA, assisting)

Algorithms & Data Structures, Spring 2019

In groups:

Implement a prioritized queueing system for an airport. You can use any priority queue algorithm, but you must be able to argue that the time complexity is no worse than $\mathcal{O}(\log N)$ for enqueue and dequeue respectively.

You should implement the priority queue in a setup that simulates passengers arriving to an airport, and passengers passing security. So you must think about the frequency of passengers in an airport, and how many are rejected (or delayed) in security – and things like that, to simulate the queues populating. So; like; random, time-of-day, ...

Passenger priority is assumed to be derived from properties like passenger category and arrival time (you decide the priority...):

1. Monkey
2. Late to flight
3. Disabled
4. Business class
5. Family

A template for such a setup can be found here:

[Airport Queue Template](#)

You may use the template, but you are strongly encouraged to think it out for yourself first, then – perhaps – seek inspiration in the provided code template.

Create a priority queue *instead* of the `NotPrioritisingPassengerArrayQueue` used in the template. Experiment with other values for producer and consumer. Try to add more than one consumer. Upload the solution, which should be

- Your own implementation, in a `PriorityQueue` class.
- A text file describing:
 - your argumentation and choice for a solution
 - your results, demonstrating that you stay $\mathcal{O}(\log N)$ for de- and en-queueing.

NB: do not upload the rest of the template, it will just confuse the reviewing process, and make extra work.

Upload to the [Peergrade website](#), no later than Tuesday March 12th, 08:30.

Please ask if you are in doubt about any of this.