

RandomQueue Report

by Simon Miller, Susan Angwec Go-Deos and Giulia Consonni

Our program `RandomQueue.java` implements the complete API and adds the extra method `resize()`.
The submission at time 18-02-2018 17:23 passed all tests on codeJudge.

The items are stored in array `Item[] listOfItem` in left-to-right order*.

The `enqueue()` operation respects this order by adding items to the right side.

The FIFO algorithm achieves this in constant time.

The `sample()` operation uses `StdRandom.uniform(0, N)` to return a random number `r` (between 0 and `N`) in constant time.

The `dequeue()` operation does picks a random index `r`, and swaps the item in index `r` with the last item of the queue [`N-1`]. Hereafter the queue is resized by one and the random item is returned.

To avoid loitering we set the last `listOfItem[N]` to null.

In our implementation, the iterator uses a temporary copy of the original array called `listOfItemClone`.
Initialising the `listOfItemClone` in the iterator takes linear time, advancing the iterator checks each item which takes constant time.

The order in which the elements are reported is uniformly chosen among all possible permutations by `StdRandom.uniform(0, N)`.

**Note that the variable names in our program differs from the ones given in the assignment and in our algorithms book.*

The known `a[]` and `N` is instead `listOfItem[]` and `amountOfItems` respectively.