Randomqueue Report

Dominico Villani, Zander Havgaard and Thomas Høpfner-Dahl
February 17, 2019

Implementation of Randomqueue

Our program RandomQueue.java implements the complete API given to us in the exercise description and passed all tests on the code Judge at time [17-02-2019 18:40].

The items are stored in a Array holding the type Item, acting as a non-FIFO queue.
The enqueue() operation adds a new Item at the first available index by keeping track of the number of items from the start of the array.
The enque() method achieves this in constant time, except for the case in which the array will be dynamically resized to fit more elements. The resizing of the array is done if the array is filled, in which case its size will be doubled. The sample operation uses the StdRandom.uniform() to get a random index for accessing an item that exists in constant time.
The dequeue operation implements a internal swap algorithm that stores the random chosen value locally, using SdtRandom.uniform(), to then swap this with index equal to numberOfItems field's value , which is the last element before null values are reached. This ensures a data structure that is not broken up by null values but can dequeue at random. Further the dequeue has the responsibility of decreasing the queue whenever the size of the Array divided by 4 is equal to the number of items in the queue. This is evaluated after every dequeue operation which ensures that a deque is done in constant time.
In our implementation, the iterator uses an integer that keeps track of the number of items left in the iterator. This integer is decremented for each next(), in which the randomly accessed item is swapped for the last item, which is then no longer accessible as the number of accessible items is decremented, until reaching zero, at which point hasnext() will return false. Initialising the array holding the queue takes linear time, advancing the iterator does one next() call which takes constant time.

The order in which the elements are reported is uniformly chosen among all possible permutations by using the StdRandom.uniform(n) method, which will a uniformly random integer between 0 and n, excluding n..