Dennis Thinh Tan Nguyen
William Diedrichsen Marstrand
Algorithms and Data Structures (Spring 2015) (BADS-SPRING 2015)

Random Queue - report

Data Structure and implemented methods

Our Random queue data structure consist of a generic array that supports insertion, deletion of a uniformly random element, and iteration in random order. One of our primary goals was to implement the enqueue method which simply added an element to the array on N + 1, where N is the number of elements within the array. Whenever N is equal to the length of the array, the length of the array is multiplied by two as a buffer which function was to reduce the occurrence of array expansion.

In the same way we implemented the dequeue method by reducing the array to half the size only when N is ¼ the size of the array's length, which also provides a buffer since we are then not forced to half or double the array size when an element is enqueued or dequeued around the middle index of the array. Furthermore the main purpose of dequeue and sample method, was to return a random element from the array.
By utilising a method called StdRandom.uniform and N numbers of elements queued up in the array, we could uniformly and independently return a random element from the array.

The difference between dequeue and sample was in fact that dequeue would remove the element retrieved from the array, while in the other hand sample would just uniformly and independently return a random element without removing it.

The iterator was simply an implementation of iterator's interface where we overwrote next(), hasNext() and added a single method called shuffle(). The interesting part is our shuffle method, which will shuffle our array, by uniformly choose a random index and switch its element with another element of an index whenever an iterator object gets created.Thus the array will randomly be shuffled and returned.

Test results

When we look at our test results we can see that each die rolls returns a random number and we can also see that the calculated mean is around 3.5 and the standard deviation is around 1.7, which was also the expected values.

The test results also indicates that the iterator shuffles the array and that each iterator has a different shuffle which is shown by the integers removed and the output of colors.

Refer to testGroupW.txt for our test results