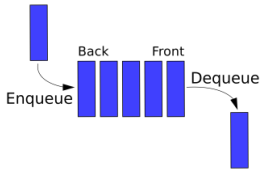
Sieve of Eratos



The sieve of Eratosthenes is one of the most efficient ways to find all primes smaller than n when n is smaller than 10 million.

1. Create a class **SieveRunner.java** with a `main` method and a class **SieveOfEratos.java** that will define the method below.

2. Write a method that returns all the prime numbers up to some integer `n` using an algorithm known as the "Sieve of Eratosthenes", which was developed by a Greek (Eratosthenes!) who lived in the third century BC. Wiki it first to see how it works. Pseudo-code:

   Fill a queue with the consecutive integers 2 through `n` inclusive.
   (Throw an IlleagalArgumentException if n < 2)
   Create a queue that will store the primes.
   While queue is not empty...
   - *Add the first value in the queue (P, the next prime) to the queue of primes.*
   - *Iterate through the queue, eliminating numbers divisible by the P that can't be primes while leaving numbers not divisible by P*
   - *if P equals √P then add remaining numbers to the queue of primes (optional, but helps with efficiency)*

```
121 >>> [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113]
```