Exercise 3.1

Question 1

- The `counts` field must be final to ensure that it will not get modified.
- The `increment` method must be synchronized to ensure atomicity.
- The `getCount` method must be synchronized to ensure that we do not read stale values.
- The `getSpan` method does not need to be synchronized as the `counts` field is final and thus cannot be modified ensuring that no stale lenght value is ever read.

Question 3

We can remove synchronized from the `increment` and `getCount` methods as the AtomicInteger class ensures that no threads will read stale values.

Question 5

Histogram2: We make the `getBuckets` method synchronized and thus retrieves a fixed snapshot of the histogram as no other operations can be done concurrently.

Histogram3: It is not possible to give a fixed snapshot without locking the operations in the `increment` method.

Histogram4: While copying the AtomicIntegerArray we use the lock of the AtomicIntegerArray to ensure that we return a fixed snapshot.