

Eli Schmitter  
November 26, 2018

## 1 R16.3

The big-Oh efficiency replacing all negative values in a linked list of Integer objects with zeroes is  $O(n)$ .  $O(n)$  is also the big-Oh of removing instead of replacing.

## 2 R16.10

when you are using an iterator and calls the removeFirst method. the current node's predecessor is no longer known making the node not easily removable.

## 3 R16.20

the big-Oh efficiency of moving the element at the head of the queue to the tail is  $O(n)$  same with moving the tail to the head.

## 4 R16.23

- $1_{\text{head}}, 2, 3, 4, 5_{\text{tail}}, 0, 0, 0, 0, 0$
- $1, 2, 3, 4_{\text{head}}, 5_{\text{tail}}, 0, 0, 0, 0, 0$
- $6, 7, 8, 9, 10_{\text{tail}}, 4_{\text{head}}, 5, 1, 2, 3, 4, 5$
- $6, 7_{\text{head}}, 8, 9, 10_{\text{tail}}, 4, 5, 1, 2, 3, 4, 5$

## 5 R16.24

To use two stacks as a queue, one is active and the other is passive. To add an object to the queue the objects stored are popped off the active stack and pushed to the passive stack. Then the new object is pushed to the active stack and then the object from the passive stack is popped and pushed back on the active stack. To remove an object from the queue pop and return from the active stack. To find the size of the queue, find the size of the of the active stack. The big-Oh running time for each of the is  $O(n)$  for adding to the stack and  $O(1)$  for removing and finding size.

## 6 R16.25

To use two queues as a stack, when adding an object to the stack it is added to an empty queue then all the other objects in the other queue are put in the queue. To remove a item from the stack you just remove the item from the nonzero size queue. to find the size find the size of the nonzero queue. The big-Oh running time for each of the is  $O(n)$  for adding to the queue and  $O(1)$  for removing and finding size.

## 7 R16.26

There is no direct way to get the unique ID from the key. But if you use a hash you can get the hash from the key.