Homework #4 – Written Questions

In class, we discussed how we could speed up the PriceQueue by adding a TreeMap field to the PriceQueue with the following invariant:

* The keys of the map are exactly the prices in the queue
* Every price in the map is mapped to the node immediately before it in the PriceQueue (unless it is the first node in the queue, in which case it doesn’t have a processor and so the price is mapped to null.

Please answer the following questions which ask you to explain how the map speeds up the various PriceQueue methods (if at all) as well as how you maintain the invariant when these methods are invoked. When explaining the changes you make to the map, please use a bullet list to help organize your answer if there are multiple changes that you make.

1. **(1 point)** How (if at all) do you use the map to speed up the delete operation?

My implementation looks up the value of the queue as a key in my TreeMap<Key, Node> instead of searching. Successful get() call returns the previous node of current key within Queue. Deletion of key is trivial.

1. **(2 points)** What are all the changes (if any) that you make to the contents of the map when a delete is performed?  
     
   To remove from TreeMap, I strictly get the first key above the key that is to be removed using higherKey(). I then get the previous node of the key to be removed using get(). Finally I put the previous node into the higher key and call Remove(key).
2. **(1 point)** How (if at all) do you use the map to speed up the enqueue operation?  
     
   To speed up the enqueue process, I checked if TreeMap contained the key and returned false if null was returned.
3. **(2 points)** What are all the changes (if any) that you make to the contents of the map when an enqueue is performed?

In enqueue, I put the price and a reference to current price’s previous node from the queue into the TreeMap<Price, Node>.

1. **(1 point)** How (if at all) do you use the map to speed up the dequeue operation?  
   I do not speed up dequeue operation. I believe it slows down because of the remove operation.
2. **(2 points)** What are all the changes (if any) that you make to the contents of the map when a dequeue is performed?

I remove the first item of the map when dequeue is called. I am unaware of what happens the reference to 1 in the second node, but it works.