Problem 2B

1. This implementation achieves O(1) time for remove() by making the minimum a private value, and storing it as it finds as we add values to the queue. It stores the difference between the added value and the minimum into the queue. If a new minimum value is added, the value stored is negative and the minimum value is updated.

**public** **void** add(**int** a)

{

**if**(totalNodes == 0)

{

min = a;

}

Node temp = head;

head.data = a - min;

head.next = temp;

temp.next = head;

**if**(totalNodes == 0)

{

tail = head;

}

**if**(head.data < 0)

{

min = a;

}

totalNodes++;

}

2. This implementation achieves O(N) time for add() by adding to the head instead of adding to the tail.

**public** **void** add(**int** a)

{

**if**(totalNodes == 0)

{

min = a;

}

Node temp = head;

head.data = a - min;

head.next = temp;

temp.next = head;

**if**(totalNodes == 0)

{

tail = head;

}

**if**(head.data < 0)

{

min = a;

}

totalNodes++;

}