

Samuel Lee
ID# 20412795

stackLL.h file

For the initial declarations such as private class, node(int x), none of the variable declarations exceed $O(1)$ runtime, so the total runtime for the private class is $O(1)$.

The following statement in the insertAt method is $O(n)$:

```
for (int c = 0; c <= i + 1; c++)
```

As we go from 0 to input i.

The rest of insertAt does not have anything above $O(1)$ runtime, so the total runtime for insertAt is $O(n)$.

The total runtime for stackLL() is $O(1)$. It is just an assignment of top.

The total runtime for ~stackLL() is $O(n)$ as it loops through the entirety of top.

```
top = top->past;
```

So we go through the list and delete each node.

The runtime of empty()

is simply $O(1)$ as it checks for true or false.

The runtime of push() is also $O(1)$ as only variable assignments/declarations are made.

The runtime of pop() is also $O(1)$ as only variable assignments/declarations, and a single return statement are made.

The total runtime is $O(1)+O(n)+O(1)+O(1)+O(n)+O(1)+O(1)+O(1) = O(n)$

queueLL.h

For the initial declarations such as private class, node(T x), none of the variable declarations exceed $O(1)$ runtime, so the total runtime for the private class is $O(1)$.

queueLL() is $O(1)$ as it just assigns nodes to nullptr.

~queueLL() is $O(n)$ as it just deletes each node in the list.

enqueue(T x) is also $O(1)$ as no loops occur.

dequeue(T x) is also $O(1)$ as no loops occur.

empty() just returns a boolean and is thus $O(1)$.

Decimate requires looping through the queue and deleting the tenth value of each, and is thus $O(n)$.

The total runtime is $O(1)+O(n)+O(1)+O(1)+O(n)+O(1)+O(1)+O(1) = O(n)$

priorityQueueLL.h

For the initial declarations such as past, future, none of the variable declarations exceed $O(1)$ runtime, so the total runtime for the private class is $O(1)$.

priorityQueueLL() just assigns two variables and is thus $O(1)$.

~priorityQueueLL() assigns and deletes variables in a loop and is thus $O(n)$.

empty() returns a boolean and is thus $O(1)$.

insert(T x) does not loop through the entire linked list and is therefore $O(1)$. The statement `if (x < h->dtype)`

Is a comparison, not a loop.

extractMin() just assigns variables and is therefore $O(1)$.

The total runtime of priorityQueueLL.h is $O(1)+O(1)+O(n)+O(1)+O(1)+O(1) = O(n)$.