CSCI 104L Lecture 13: Exceptions and Sorting

What would be the correct way of handling the following request?

```cpp
LinkedList<int> *LL = new LinkedList;
for (int i = 0; i < 10; i++) LL->prepend(i);
cout << LL->get(15) << endl;
```

Put this in LinkedList.cpp:

```cpp
#include<exception>
#include<stdexcept>

... if (position >= this->size()) throw logic_error("position was too large!");
```

A thrown exception will propagate up through the program stack until it reaches a piece of code designed to handle it. If no such code is found, the program terminates.

The user should do this:

```cpp
try {
    cout << LL->get(15) << endl;
    cout << "Printed successfully!" << endl;
} catch (logic_error &e) {
    cout << "A logic error occurred!" << endl;
    cout << e.what();
} catch (exception &e) {
    cout << "General exception" << endl;
}
```

Sorting

Given an array `a`, rearrange the items so that they are in non-decreasing order. That is, `a[i+1] ≥ a[i], ∀i : 0 ≤ i ≤ size - 2`. The sorted array must have the same elements that were in the original array.

**Bubble Sort**

```cpp
for (int i = n-1; i > 0; i--) {
    for (int j = 0; j < i; j++) {
        if (a[j] > a[j+1]) a.swap(j, j+1);
    }
}
```

A sorting algorithm is **stable** if all elements with the same value have the same relative ordering as before the sorting.

- Is Bubble Sort stable?
- What kind of proof would you use to prove Bubble Sort works?
- What is the runtime of Bubble Sort?