CSCI 104L Lecture 12: Error Handling and Interpolation Search

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

```
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:

```
#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:

```
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;
}
```

### Interpolation Search

Let `a`:

```
int InterpolationSearch(T x, int l, int r) {
    if (r < l) return -1;
    if (a[r] == a[l]) {
        if (x == a[r]) return r;
        return -1;
    }
    int m = (x-a[l])/(a[r]-a[l])*(r-l)+l;
    if (a[m] == x) return m;
    if (a[m] < x) return InterpolationSearch(x, m+1, r);
    return InterpolationSearch(x, l, m-1);
}
```

To get a good result, we are assuming that the elements of the array are “roughly evenly spread.” If you assume this is true (for a precise mathematical definition of this statement, which we are not providing) then you would find an average case runtime of \(O(\log \log n)\).