CMPT 225 Lab 10: Element Uniqueness

The problem

Element Uniqueness asks the following question: Given a sequence S of n numbers, are the elements of the sequence all unique?

The code

You are to implement a C++ function to answer the element uniqueness question and code to test this procedure. You are given no starter code; successful completion of this lab requires you to create all of the code structure specified.

First, there will be a class called Unique which has the following public function:

```
bool unique(int* A, int n);     // returns true if all n elements in the
```

You should implement unique in an efficient O(n log n) manner. You may not use STL or other external libraries.

Second, there will be a separate class called Test with a public function

```
void test(int n, int max);      // creates a new array of n random integ
                           // then runs unique on this array and prints
```

To create a random number, you must use **#include <cstdlib>** and **#include <time.h>.** The protocol for use is to call

```
    srand(time(NULL));
```

once at the beginning of your program (in main() will do) and then call **rand()** every time you need a random number. **rand()** returns a random integer, which you will need to truncate to the range [0, max) by using the remainder operator **%**:

```
    int number = rand() % max;
```

Your **main** function (which you can include in Test.cpp) should call test(5, 10), test(20, 100), and test(50, 200).

That completes the program. Run the program several times until you are sure that it is correct (you need to see both the true and false cases of unique). Once you are convinced that it works, call the TA to witness your results and get your mark for this lab.
