COLBY COLLEGE
Department of Mathematics
and Computer Science

CS115 **Allen B. Downey**
Spring 1998

Assignment 11: Arrays

Due: Wednesday 28 April

Random integers

The book shows examples of how to use the `Math.random` method to generate random `double`s between 0.0 and 1.0. By scaling and shifting these values, you can generate random values in a given range. For example, `Math.random() * 100.0 + 10.0` yields a `double` between 10 and 110.

1. Write a method called `randInt` that takes two arguments, `low` and `high`, and that returns a random integer between low and high.
2. Write a method called `randIntArray` that is similar to `randomArray` on page 115, except it should take two additional arguments, `low` and `high`, and it should return an array of integers in the range from `low` to `high`.
3. Write a method called `prIntArray` that takes an array of integers and prints them. Use it to debug `randIntArray`.

`for` loops

1. Using a `for` loop, write a method called `inRange` that counts the number of elements in an integer array that fall in a given range (from `low` to `high`, including both ends).
2. As an exercise, you may want to rewrite the methods on Quiz 15 using `for` loops. This step is not required, but it will be excellent practice for the final exam.

Recursion

It's like a nightmare that doesn't go away when you wake up. Many of the patterns we have seen for traversing arrays can also be written recursively. It is not common to do so, but it is a useful exercise.

1. Write a method called `maxInRange` that takes an array of integers and a range of indices (`low` and `high`), and that finds the maximum value in the array, considering only the elements between `low` and `high`, including both ends.

   This method should be recursive. If the length of the range is 1, that is, if `low == high`, we know immediately that the sole element in the range must be the maximum. So that's the base case. If the array is bigger than 1, we can break the array into two pieces, find the maximum in each of the pieces, and then find the maximum of each of the piece-maxima.

2. One you have `maxInRange` working, you can write a *wrapper* method called `max` that works like this

```
public static int max (int[] a) {
  return maxInRange (a, 0, a.length-1);
}
```

   The wrapper method uses the `worker` method to do the real work. The role of the wrapper is to provide an interface to the outside world that is easier to use.

3. Write a recursive version of `find` using the wrapper-worker pattern. `find` should take an array of integers and a target integer. It should return the index of the first location where the target integer appears in the array, or -1 if it does not appear. `find` was called `grapefruit` on the exam.

Sorting

One not-very-efficient way to sort the elements of an array is to find the largest element and swap it with the first element, then find the second-largest element and swap it with the second, and so on.

1. Write a method called `indexOfMaxInRange` that finds the largest element in the given range, and returns *its index*. You can modify your recursive version of `maxInRange` or you can write an iterative version from scratch.

2. Write a method called `swapElement` that takes an array of integers and two indices, and that swaps the elements at the given indices.

3. Write a method called `sortArray` that takes an array of integers and that uses `indexOfMaxInRange` and `swapElement` to sort the array from largest to smallest.