

Chapter 7 Problems: Part 1

11. Write a method called `isUnique` that accepts an array of integers as a parameter and returns a `boolean` value indicating whether or not the values in the array are unique (`true` for yes, `false` for no). The values in the list are considered unique if there is no pair of values that are equal. For example, if passed an array containing `[3,8,12,2,9,17,43,-8,46]`, your method should return `true`, but if passed `[4,7,3,9,12,-47,3,74]`, your method should return `false` because the value 3 appears twice.

```
public static boolean isUnique(int[] arr) {
    Arrays.sort(arr);
    for (int i = 0; i < arr.length; i++) {
        if (arr[i+1] == arr[i]) {
            return false;
        }
        else if (arr[i] == arr.length - 1){
            return true;
        }
        else {
            ;
        }
    }
    return true;
}
```

12. Write a method called `priceIsRight` that mimics the guessing rules from the game show *The Price Is Right*. The method accepts as parameters an array of integers representing the contestants' bids and an integer representing a correct price. The method returns the element in the bids array that is closest in value to the correct price without being larger than that price. For example, if an array called `bids` stores the values `[200,300,250,1,950,40]`, the call of `priceIsRight(bids, 280)` should return `250`, since 250 is the bid closest to 280 without going over 280. If all bids are larger than the correct price, your method should return -1.

```
public static int priceIsRight(int[] arr, int target) {
    Arrays.sort(arr);
    for (int i = 0; i < arr.length; i++) {
        if (target > arr[i]) {
            ;
        }
    }
}
```

```

        else if (target < arr[0]) {
            return -1;
        }
        else if (target < arr[i]) {
            return arr[i-1];
        }
    }
    return -1;
}

```

13. Write a method called `longestSortedSequence` that accepts an array of integers as a parameter and returns the length of the longest sorted (nondecreasing) sequence of integers in the array. For example, in the array `[3,8,10,1,9,14,-3,0,14,207,56,98,12]`, the longest sorted sequence in the array has four values in it (the sequence `-3,0,14,207`), so your method would return 4 if passed this array. Sorted means nondecreasing, so a sequence could contain duplicates. Your method should return 0 if passed an empty array.

```

public static int longestSortedSequence(int[] arr) {
    int maxCount = 1;
    int count = 1;
    for (int i = 1; i < arr.length; i++) {
        if (arr[i-1] ≤ arr[i]) {
            count++;
        }
        else {
            if(count > maxCount) {
                maxCount = count;
            }
            count = 1;
        }
    }
    return Math.max(maxCount, count);
}

```