

CS2302 - Data Structures

Spring 2018

Lab 2

Due Monday, February 12

For this lab you will implement several methods to determine if an array of integers in a certain range contains duplicate elements. First, you need to write a method that receives two integers n and m and builds and returns an array of length n that contains random integers in the 0 to $m - 1$ range. Then implement the following methods to determine if the array contains duplicates:

1. Compare every element in the array with every other element in the array using nested loops as follows:

```
public static boolean hasDuplicates(int []A){
    boolean duplicates = false;
    for (int i=0;i<A.length;i++)
        for (int j=0;j<A.length;j++)
            if (A[i]==A[j] && i!=j)
                duplicates = true;
    return duplicates;
}
```

2. In the previous method, we compare every pair of elements twice and we also compare every element with itself, which is unnecessary. Also, once a duplicate is found, the program continues making comparisons, which is also unnecessary. Improve the code by eliminating these inefficiencies.
3. Note that if the array is sorted, identical elements must occupy consecutive positions in the array. Write a method that sorts the array using selection sort, then finds duplicates by performing a single pass through the sorted array.
4. Repeat the previous point, but now use quicksort to sort the array.
5. We can take advantage of the fact that the range of the numbers in the array is fixed (0 to $m - 1$). Use a Boolean array S of length m to indicate if elements in the array have been seen before. Then determine if there are duplicates by performing a single pass through the unsorted array.

Determine the big-O running time of each of the previous methods. Run experiments with several different values of n and m (with $m < n$) and determine the running times with respect to n and illustrate this by means of plots or tables. Discuss whether the theoretical results match the outcomes of the experiments, making sure you evaluate the algorithms' behavior with a wide range of values of n . Write a report discussing your work, as described in the syllabus.