UNIVERSITY OF VIRGINIA, DEPARTMENT OF COMPUTER SCIENCE

Big Oh - Analyzing Scalability of Algorithms

Nada Basit and Mark Floryan

March 14, 2022

1 SUMMARY

For this homework, you will implement a few simple methods. Each of these methods is expected to run at a different *asymptotic complexity*, including $\Theta(n)$, $\Theta(\log n)$, $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3)$, and $\Theta(2^n)$. You will focus on implementing the methods in this homework, and analyze their time complexity for the analysis homework (the next homework).

1. Download the starter code and import the project into Eclipse.
2. Implement the missing methods from BigOh.java
3. Verify the correctness of your algorithms and submit to Gradescope to check.
4. **FILES TO DOWNLOAD:** BigOh.zip
5. **FILES TO SUBMIT:** BigOh.java

1.1 DOWNLOAD STARTER CODE AND IMPLEMENT FIVE METHODS

You can download the starter code for this homework from the course repository here. Once you have done so, you should import the project into Eclipse.

This project contains two Java files. *BigOh.java* contains the methods you need to implement and *Main.java* contains a main function we have written to help you test your code and measure the time each method takes to execute. *BigOh.java* contains the following methods (most of which are not yet implemented):

```
/* Binary Search: Should run in Theta(logn) time */
/* Returns true if item is in the array a */
public static binarySearch(int[] a, int item);

/* Max value in array: Should run in Theta(n) time */
public static int max(int[] a);

/* Calls binary search n times. Counts number of successful searches */
/* You should search for the numbers 1 through n in succession */
/* Should run in Theta(nlogn) time */
public static int multipleBinarySearch(int[] a);

/* Counts pairs of numbers whose sum is multiple of 5 */
/* Should run in Theta(n^2) time */
public static int allPairs(int[] a);

/* Counts the pairs of three in the list a,b,c in which a+b=c */
/* Should run in Theta(n^3) time */
public static int allTriads(int[] a);

/* Prints all subsets of a */
/* Should run in Theta(2^n) time */
/* e..g, {1,2,3} would print {},{1},{2},{3},{1,2},{1,3},{2,3},{1,2,3} */
public static int allSubsets(int[] a);
```

Your first task is to implement the first five of these methods. The allSubsets() method is provided as it is much more tricky than the others.

Many of you have probably not seen **binary search** before (we may go over this algorithm in lecture if we have time). *Binary search* is an efficient method of searching through sorted data. Because the data is sorted, we can efficiently search by looking in the middle of the data and reducing our search to the half that is guaranteed to contain the item. We do this continuously until the search narrows all the way down to one item. This is similar to looking up a word efficiently in a dictionary. First, we look in the middle of the book. If our word is in the lower half we open halfway in that direction, and so on until we narrow in on the word we are searching for. Some psuedo-code for binary search can be found below:

```
binarySearch(int[] a, int item){
    set variables min=0 and max=a.length-1
    continue while min is less than or equal to max
```

```
4           middle = middle element between min and max
            if middle element is item, then found!
6           if item is less than middle element
                repeat search between min and middle-1
8           if item is more than middle element
                repeat search between middle+1 and max
10
        if this loop exits, then the item was not found
12  }
```

The other methods should be self-explanatory from the comments above. Let course staff know if you have any confusion about these. Make sure to use the provided main method to test your code. We are not providing tests for this assignment.

1.2 GRADESCOPE

You should submit your code to *Gradescope*. If you are having trouble with your submission, you should double check the following common problems:

1. Make sure you are only submitting one file, and it is called *BigOh.java* exactly.

2. Make sure there is keep the package statement at the top of the file (package main;). You can remove the one that is given in the default file that is provided when you submit.

3. Your code should **NOT contain any additional import statements** **the import of Vector can remain.*

4. Make sure you **DO NOT** submit a *main method* in your file.

5. Make sure your file is **NOT printing anything to the console**. Your methods should simply return the correct results.