Algorithm Complexity

Programming Projects:

1) Max Number Finder

- Start Eclipse and create a new Java project called ‘lab4’
- Create class called **NumberChecker**
- Add a method with the following signature –

```
int findMax(Integer [] numbers)
```

- Now implement this method, so that it returns the largest number within the given **numbers** array (hint you will need a loop and variable called max)
- Write a Driver class which creates an instance of **NumberChecker** then calls the findMax() in order to test. e.g.

```
NumberChecker numChecker = new NumberChecker ();
int max = numChecker.findMax(new Integer[] {5,2,7,9,10,1,2});
System.out.println(“Max number is “ + max);
```

- Determine the ‘growth function’ and ‘order’ of the implemented method. Express the complexity using the Big-O notation.

Think you’re finished? Check you have commented your code and added Javadoc to the methods and classes. Include in the javadoc a comment about the complexity of the implemented method algorithms.

2) Number Grid Display

- Create class called **NumberGrid**
- Add a method with the following signature –

  ```
  void output(int range);
  ```

- Now implement this method, so that it outputs a number of rows and columns (hint use a nested loop) as determined by the given range value. Each row should display the current row number (*range* times). e.g. if called with the number 2 the output should be –

  ```
  1    1
  2    2
  ```

- Whereas, if called with the number 5 the output should be –

  ```
  1    1    1    1    1
  2    2    2    2    2
  3    3    3    3    3
  4    4    4    4    4
  5    5    5    5    5
  ```

- Determine the ‘growth function’ and ‘order’ of the implemented method. Express the complexity using the Big-O notation.

Think you’re finished? Check you have commented your code and added Javadoc to the methods and classes. Include in the javadoc a comment about the complexity of the implemented method algorithms.