CS 112 Spring 2019– Homework Four
Due March 25th at 11:59pm using `OneDrive`
No late assignments will be accepted.

1 PartA

You will write your answers to partA in a text file called hw4.txt. Make sure to submit this file back to us using your OneDrive.

1. By using the definition of BigO i.e., $\mathcal{O}(...)$, show that:
   (a) $6n^2 + 3$ is $\mathcal{O}(n^2)$
   (b) $n^2 + 17n + 1$ is $\mathcal{O}(n^2)$
   (c) $5n^3 + 100n^2 - n - 10$ is $\mathcal{O}(n^3)$
   (d) $3n^2 + 2^n$ is $\mathcal{O}(2^n)$
2. Show that $7n^2 + 5n$ is not $\mathcal{O}(n)$
3. Algorithm X requires $n^2 + 9n + 5$ operations, and algorithm Y requires $5n^2$ operations. What can you conclude about the time required for these algorithms when n is small and when n is large? Which is the faster algorithm in these two cases?
4. Using the BigO, i.e., $\mathcal{O}(...)$ notation, indicate the time requirement of each of the following tasks in the worst case, Describe any assumptions that you make
   (a) After arriving at a party, you shake hands with each person there.
   (b) Each person in a room shakes hands with everyone else in the room.
   (c) You climb a flight of stairs
   (d) You slide down the banister
   (e) After entering an elevator, you press a button to choose a floor
   (f) You ride the elevator from the ground floor up to the nth floor
   (g) You read a book twice
5. Describe a way to climb from the bottom of a flight of stairs to the top in a time that is no better than $\mathcal{O}(n^2)$.
6. What is the $\mathcal{O}(...)$ of the following computations?

   (a)
```
int sum = 0;
for (int counter = n; counter > 0; counter = counter - 2)
   sum = sum + counter;
```

   (b)
```
int sum = 0;
for (int counter = 1; counter < n; counter = 2 * counter)
   sum = sum + counter;
```

   (c)
```
for (int pass = 1; pass <= n; pass++)
{
   for (int index = 0; index < n; index++)
   {
      for (int count = 1; count < 10; count++)
      {
         . . .
      } // end for
   } // end for
} // end for
```

(d) What is the Big-Oh of method1? Is there a best case and a worst case?

```
public static void method1(int[] array, int n)
{
   for (int index = 0; index < n - 1; index++)
   {
      int mark = privateMethod1(array, index, n  1);
      int temp = array[index];
      array[index] = array[mark];
      array[mark] = temp;
   } // end for
} // end method1
public static int privateMethod1(int[] array, int first, int last)
{
   int min = array[first];
   int indexOfMin = first;
   for (int index = first + 1; index <= last; index++)
   {
      if (array[index] < min)
      {
         min = array[index];
         indexOfMin = index;
      } // end if
   } // end for
   return indexOfMin;
} // end privateMethod1
```

(e) What is the Big-Oh of method2? Is there a best case and a worst case?

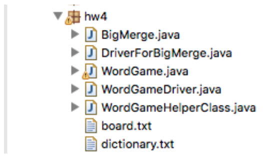
```
public static void method2(int[] array, int n)
{
   for (int index = 1; index <= n - 1; index++)
      privateMethod2(array[index], array, 0, index - 1);
} // end method2
public static void privateMethod2(int entry, int[] array, int begin, int end)
{
   int index = end;
   while ((index >= begin) && (entry < array[index]))
   {
      array[index + 1] = array[index];
      index;
   } // end while
   array[index + 1] = entry;
} // end privateMethod2
```

2 PartB

We have provided you on Blackboard the following Java files for PartB of this assignment:

- `BigMerge.java`
- `DriverForBigMerge.java`
- `WordGame.java`
- `WordGameDriver.java`
- `WordGameHelperClass.java`

You are required to write code only in `BigMerge.java`, `WordGame.java` and `WordGameHelperClass.java`. Each one of these classes, have a comment starting with `TODO` that indicates what you are suppose to do. Inside your Eclipse, you must have the following arrangement of files inside your `hw4` package before writing any code:



```
hw4
  BigMerge.java
  DriverForBigMerge.java
  WordGame.java
  WordGameDriver.java
  WordGameHelperClass.java
  board.txt
  dictionary.txt
```

Figure 1: Arrangement of hw4 files inside hw4 package

2.1 Word Game

We provide you with code for implementing the following game: given a rectangle of letters, find all dictionary words that consist of consecutive letters on the board (up-down, down-up, left-right, or right-left). We provide a dictionary file (999 most common American English words) and a sample input, as `dictionary.txt` and `board.txt` files. We also provide a correct output for this input as `example-output.txt` file. Implement the code according to instructions. Don't neglect to answer one question in the comments of the code. Your submitted code should not have any new or modified print statements (they are useful for debugging, but remove them once you are done). You can change `WordGameDriver` however you want (including adding print statements); we won't grade it.

If you are feeling ambitious, you can download other dictionaries – for example, the huge (178,690 words) `https://www.wordgamedictionary.com/twl06/download/twl06.txt` used for Scrabble. That way your program will find more words. However, please don't submit huge dictionaries with your assignment.

Note: We have provided you with lots of comments in the code to help you get started. All methods that have a **TODO** as part of comments are methods that you MUST complete. Do **not** change the method signature or the name of the class in any way. This will break our tests and will result in you getting 0 for this question.

2.2 Merge Sorted Arrays

The most important procedure in merge sort takes two sorted arrays and merges them into one. Implement a method that takes many sorted arrays—in fact, a whole array of sorted arrays—and, similarly, merges them into one. The process should be essentially the same: find the minimum, move it over. We have to be careful in finding the minimum when some of the arrays run out of elements. First, implement the method `argMin` that finds the minimum. You are asked only to edit the file `BigMerge .java`. As always, do not change file names or method names. We have provided a class called `DriverForBigMerge.java` that contains the `main` function. You will use this class to run your code. Do not add a `main` function inside `BigMerge.java`.

Note: We have provided you with lots of comments in the code to help you get started. All methods that have a **TODO** as part of comments are methods that you MUST complete. Do **not** change the method signature or the name of the class in any way. This will break our tests and will result in you getting 0 for this question.

3 Submission Checklist[1]

Similar to your previous assignment, you will create a new package `hw4` in Eclipse. Inside this package, all your Java files will reside. You will only submit the following files from the package `hw4` to us using OneDrive. I have already created the directory `hw4` for you, and please do not recreate the `hw4` directory. All you are asked to do is place the following Java files, inside the `hw4` directory of your OneDrive. This package must contain the following files only. We do not require your driver files, so please make sure to remove these.

- `BigMerge.java`
- `WordGame.java`
- `WordGameHelperClass.java`
- `hw4.txt`

The first line in every Java file must be `package hw4;`
Be sure to do the following:

- You have read the Java Style Guide for CS112 (linked on BlackBoard), and followed all guidelines regarding format, layout, spaces, blank lines, and comments;
- You have removed or changed all comments inherited from my templates which do not relate to your solution;
- You have verified that your program satisfies all the performance tests in the templates.

You can add as many helper methods (these must all be `private`); however, YOU CANNOT CHANGE THE METHOD NAMES OR CHANGE THE SIGNATURE OF THE REQUIRED METHODS IN THIS ASSIGNMENT. THIS WILL BREAK OUR TESTS, AND YOU WILL RECEIVE ZERO.

[1] If you deviate from these instructions, we will penalize you 30% on the entire assignment. No late work will be accepted