COP 3530 – Data Structures
Homework 1
Full grade: 100 points

Please read carefully the homework-solving guidelines from the syllabus before you start. Plagiarism is absolutely unacceptable.

1 Problem description

Online news portals are always interested in analyzing what triggers high user visits. Say you work as a data analyst for such an online portal. Out of the many tasks you perform for them, you occasionally figure out the longest profitable periods every financial year when many new users visit the portal. Such periods could range over minutes, hours, days, and even months.

The portal at your company is smart. At the end of every minute, it logs the number of new users. At every minute i, it generates a new integer k_i that denotes the user activity during the minute. Let a_i be the number of new users and b_i be the number of old users who visited the portal from their cellphones or computers during minute i. For this portal, k_i is defined as $b - a$. Clearly, positive k_is are highly desired by the company. It would mean the portal is gaining online popularity.

As an analyst, you get an array A of length n containing n number of k_i values generated by the portal for a specific period. Your task is to find out the sub-array of A whose sum is the maximum possible. The company considers it to be the longest profitable period. In some cases (for instance, when every integer is positive), that sub-array could be the array A itself. Note that the items in a sub-array of A are also contiguous in A. Here is an example for you for $n = 11$. Let A equals $[-20, -40, 80, -10, 50, 90, -70, -20, 40, -30, 20]$. In this case, the required sub-array is $[80, -10, 50, 90]$ (starts at index 2 and ends at index 5) and its sum is 210 (the maximum among all sub-arrays of A, including A itself). Assume that it cannot be the case that every integer in A is zero. So, arrays such as $[0, 0, 0, 0]$ are considered to be invalid inputs.

You first task is to complete the $O(n^3)$-time brute-force method **`enCubedImplementation`** inside the class **`PortalTrafficAnalyzer`**.

Your next task is to complete the method named **`enSquaredImplementation`** inside the class **`PortalTrafficAnalyzer`** by implementing an algorithm that runs in $O(n^2)$. You ***are not allowed*** to declare any new array inside **`enSquaredImplementation`**. This method is expected to be substantially faster than the **`enCubedImplementation`** method. For instance, when $n = 5K$, one can observe around $1500X$ or even more speedup. See the supplied tester class named **`TestPortalTrafficAnalyzer`** to see how to observe speedup.

2 What to deliver?

Do not use any built-in data structure from Java Collections in this homework. See here for those classes: `https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/doc-files/coll-overview.html`.

Upload just the file **`PortalTrafficAnalyzer.java`** to Canvas. Feel free to define additional helper methods wherever needed. No need to upload anything else. ***Further, copy-paste your console output to the top of your java file so we can see the speedups obtained on your computer. For sample console output, see the main method of the tester class.***

Commenting is highly encouraged. Please test thoroughly before submission. You will be given zero if your program does not compile or gives unnecessary warnings.

You are allowed to submit your solutions multiple times. We will grade your latest submission only.