ITU Computer Engineering Department
BLG 335E Analysis of Algorithms I, Fall 2022
Project #2
Due December 2, 2022 11:59pm

Problem Definition

In an online data stream, statistics of the data hold the key that can provide some intuition about the general trends and even have some predictions regarding the future. In this regard, the estimations of the statistics on the large data pools should be as swift and precise as possible.

In this homework, we are aiming to find out the mean, standard deviation, and 5-figure summary of a household electric power consumption[1] data stream for each of the requests that have been made in an online manner. In particular, a 5-figure summary consists of a minimum, 1^{st} quartile, median, 3^{rd} quartile, and maximum element. To make it happen, it is crucial to store and process the incoming data in a time-efficient manner.

To calculate the statistics of the online data stream efficiently, we ask you to choose appropriate data structures and implement their methods. Afterwards, we would like you to adopt these data structures for this problem.

Constraints

- The maximum number of samples is denoted by N, where $1 \leq N \leq 1000000$.
- At least two samples have been provided before estimating any of the mean, standard deviation, or 5-figure summary statistics.
- **WARNING**: Your program runtime is subject to a timeout, you should be writing it as efficient as possible. Try to remember the data structures as array, binary (search) tree, and heap to improve the efficiency of your code.

Sample Cases

In the first line of the test case, we provide the total number of statistic estimators, and we list their names down below. As it can be seen in input1.txt, there are 7 estimators in total which are **mean** (for average estimation), **std** (for standard deviation estimation), **min** (for minimum element estimation), **firstq** (for the first quantile estimation), **median** (for median element estimation), **thirdq** (for the third quantile estimation) and **max** (for maximum element estimation). After that, we leave one empty line, and state how many lines are present for the rest of the test case and which feature of the dataset that we are interested in, separated by a comma. After that, it is possible to proceed either with one of the following. First, we can append a sample by writing "*add*" (as in line 11) and then, the features of the data sample in the line below (as in line 12). Here, we enter the day, time, global active power (**gap**), global reactive power (**grp**), voltage (**v**) and global intensity (**gi**) features which are separated with commas. Second, it is possible to flush the statistics by writing "*print*". When the print option is called, we should see the starting date, starting time, latest retrieved date, latest retrieved time, mean, standard deviation, minimum value, first quadrant, median, third quadrant, and maximum value in output1.txt, given that we have 7 estimators shown in input1.txt.

[1] `https://www.kaggle.com/datasets/thedevastator/240000-household-electricity-consumption-records/versions/1?resource=download#`

Code Listing 1: input1.txt

```
7
mean
std
min
firstq
median
thirdq
max

19,gap
add
1/1/07,0:00:00,2.58,0.136,241.97,10.6
add
1/1/07,0:01:00,2.552,0.1,241.75,10.4
print
add
1/1/07,0:02:00,2.55,0.1,241.64,10.4
print
add
1/1/07,0:03:00,2.55,0.1,241.71,10.4
print
add
1/1/07,0:04:00,2.554,0.1,241.98,10.4
add
1/1/07,0:05:00,2.55,0.1,241.83,10.4
print
add
1/1/07,0:06:00,2.534,0.096,241.07,10.4
print
```

Code Listing 2: output1.txt

```
1/1/07,0:00:00,1/1/07,0:01:00,2.566,0.01979899,2.552,2.559,2.566,2.573,2.580
1/1/07,0:00:00,1/1/07,0:02:00,2.560667,0.01677299,2.550,2.551,2.552,2.566,2.580
1/1/07,0:00:00,1/1/07,0:03:00,2.558,0.0146969,2.55,2.55,2.551,2.559,2.58
1/1/07,0:00:00,1/1/07,0:05:00,2.556,0.01186592,2.550,2.550,2.551,2.554,2.580
1/1/07,0:00:00,1/1/07,0:06:00,2.552857,0.01365563,2.534,2.550,2.550,2.553,2.580
```

Project Deliveries

- Your efficient C++ Implementation of online statistics estimator on household power consumption data [70 pts]
- Report on the project [30 pts]
  - BONUS: 1.333x multiplier on the report, if written with LaTeX. Please consider looking at this repository if you are seeking a template: `https://github.com/ongun-kanat/itu-report-templates`

Report Structure

Your report should explain briefly the data structures that you have used in your implementation, where you should also include the functions with their pseudocodes and asymptotic upper bounds. After that, in a separate section, you should describe how exactly you used these data structures in this problem to estimate the statistics. Similarly, you should include the functions for estimating the statistics with their pseudocodes and **calculated** complexity values.

Then, create a **line plot** by recording how long it takes to complete the data structure operations and how many times your implementation calls the data structure methods for all input text files and each of the statistics estimators. This part should be repeated 10 times. Evaluate your results briefly to support your claims on the calculated complexity values.

In a separate section, please provide an answer to the following question in your own words: Describe whether it is possible to apply a sliding window-based approach to estimate the statistics while retaining the same time complexity.

Submission Rules

- You should write your code in C++ language and try to follow an **object-oriented methodology** with well-chosen variables, methods, and class names and comments where necessary.
- You **cannot** use the C++ Standard Template Library (STL) algorithms. In addition, the use of specialized STL containers like deque, queue or priority queue is not permitted, but the use of vector or list is allowed.
- You can define multiple classes in a single cpp file or use multiple cpp files with header files.
- It is mandatory to include a MakeFile which makes your code compiled with the ”**make all**” command.
- Also, make sure that your code can be run **on our Docker container** in the form of **./homework2 [input_file]**.
- If the code is not self-explanatory and does not include adequate comments, a point penalty of up to 20 points will be applied.
- Hand-written reports will not be accepted. Also for pseudocodes, please prepare them properly, do not copy and paste your exact C++ code that you used for your assignments.
- Do not share any code or text that can be submitted as a part of an assignment (discussing ideas is okay).
- Only electronic submissions through Ninova will be accepted no later than the deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not **share or copy code** from your classmates or the Internet. You should submit your **own individual** homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- If you have any questions about the homework, you can send an e-mail to Caner Özer (ozerc@itu.edu.tr).
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**