Fall 2024 CS685/785 Foundation of Data Science

Coding Assignment 4: Data Stream

Instructor: Dr. Xi Li (xli7@uab.edu)
TA: Mengchen Fan (fanm@uab.edu), Ashraful Islam (aislam@uab.edu), and Britt Guimond (bguimond@uab.edu)

Objective

1. Reinforce theoretical concepts covered in lectures.
2. Address a question from a real coding interview related to obtaining output from real-time data streaming.

Tools Required

You may need either the SortedDict from the sortedcontainers package or a heap.

Problem Description

You are given a stream of records about a particular stock. Each record contains a timestamp and the corresponding price of the stock at that timestamp.

Unfortunately due to the volatile nature of the stock market, the records do not come in order. Even worse, some records may be incorrect. Another record with the same timestamp may appear later in the stream correcting the price of the previous wrong record.

Design an algorithm that:

- Updates the price of the stock at a particular timestamp, correcting the price from any previous records at the timestamp.
- Finds the latest price of the stock based on the current records. The latest price is the price at the latest timestamp recorded.
- Finds the maximum price the stock has been based on the current records.
- Finds the minimum price the stock has been based on the current records.

Implement the StockPrice class:

- StockPrice() Initializes the object with no price records.
- update(int timestamp, int price) Updates the price of the stock at the given timestamp.
- current() Returns the latest price of the stock.
- maximum() Returns the maximum price of the stock.
- minimum() Returns the minimum price of the stock.

Example

```
obj = StockPrice();
obj.update(1, 10); // Timestamps are [1] with corresponding prices [10].
obj.update(2, 5);  // Timestamps are [1,2] with corresponding prices [10,5].
obj.current();     // return 5, the latest timestamp is 2 with the price being 5.
obj.maximum();     // return 10, the maximum price is 10 at timestamp 1.
obj.update(1, 3);  // The previous timestamp 1 had the wrong price, so it is updated to 3.
                   // Timestamps are [1,2] with corresponding prices [3,5].
obj.maximum();     // return 5, the maximum price is 5 after the correction.
```

obj.update(4, 2); // Timestamps are [1,2,4] with corresponding prices [3,5,2].
obj.minimum();   // return 2, the minimum price is 2 at timestamp 4.

Submission

Submit the **HTML** file that includes **both** your **code** and **results** by selecting “Save and export notebook” -> “HTML”.