Build a simplified version of Google Finance.

In this version, we would like to track the price of a single stock. We want to show the highest, lowest, and most recent price of the stock. We will get a live feed from the stock exchange containing price updates (timestamp, price).

In addition to updates, the feed can contain corrections to past prices (e.g. traders mistype a price). A correction can either be (timestamp, new\_price) or (timestamp), where the latter means to drop the price from that timestamp.

Implement a class that ingests the feed so that we can show the highest, lowest, and most recent price of the stock. Assume the feed is already parsed and the caller will invoke the appropriate methods on your class.

1 2 3 5 6 0.5 12 51 4

Lowest = 0.5

Highest = 6

Latest = 0.5

Latest\_timestamp = DateTime()

1. What if the updated value is the latest timestamp. How do we identify if it’s the latest?

class GoogleFinance:

Stock\_values = None

Lowest\_value = None

Highest\_value = None

Latest\_value = None

Latest\_timestamp = None

Def \_\_init\_\_(self):

Stock\_values = dict()

Def add\_new\_value(timestamp, price):

Self.stock\_values[timestamp] = price

Self.lowest\_value = min(self.lowest\_value, price)

Self.highest\_value = max(self.highest\_value, price)

Self.latest\_timestamp = timestamp

Def update\_value(timestamp, price = None):

If price:

If timestamp == self.latest\_timestamp:

Self.latest\_value = price

Self.latest\_timestamp = timestamp

Self.stock\_values[timestamp] = price

Else:

If timestamp == self.latest\_timestamp:

Self.latest\_timestamp = self.stock\_values.keys.to\_list()[-1]

Self.latest\_value = self.stock\_values.values.to\_list()[-1]

self.stock\_values.drop(timestamp)

Self.lowest\_value = min(self.stock\_values.values)

Self.highest\_value = max(self.stock\_values.values)

A treemap