# Data Engineer coding task

## Convert Level 3 to Level 1 data.

You may write code in any language you are comfortable with. There should be clear instructions on how to execute the code. Note that this is a data engineering task and not a data science task, i.e. it's about engineering systems and not producing analytic insights.

# Task descriptions

The purpose of an exchange is to match market participants orders (buy against sell). The state of the market is represented in an **order book**. This shows for each price level and each side (buy/sell) the orders for that level.

**Level 3** (L3) data are updates to market participants' orders. Because sending the full state of the order book for each modification to the book is slow and wastes bandwidth, the exchange publishes deltas/diffs instead, which describes updates to the state of the book.

Each order has a side, price and quantity. They are uniquely identified by an **order id**. An order update can be one of:

- **ADD** inserts a new order into the book.
- **UPDATE** modifies an existing order on the book.
- **DELETE** removes the order from the book.
- **TRADE** matches buy orders with sell orders at a specific price; the corresponding update to the book being a reduction of quantity outstanding in the respective orders according to the amount traded. (separate TRADE messages will be sent for buy vs. sell, so you only need to handle one side at a time; also the other side of the trade may never show up in the book, i.e. they are executed immediately)

*The lowest sell price level (ask price)* with all the cumulative quantity at ask price level (ask size) and *the highest buy price level (bid price)* with all the cumulative quantity at bid price level (bid size) constitute what we call the **BBO** - best bid and ask (offer).

**Level 1** (L1) data refers to the stream of updates to BBO as a consequence of updates to the book state. It is derived data, with the L3 data being the raw data. Unlike with L3 data the full state of the BBO should be published each time as it is not a voluminous amount of data.

Things to consider:

- Bonus points for a streaming implementation.
- How would you deal with data which arrives out of order?
- How might it be possible for a unified batch and streaming implementation to work?

Use the provided CSV files as L3 data and expected L1 data to test your program.

# Explanation on L3 data

| Column | Description |
|---|---|
| seq_num | Sequence number of L3 market data which defines an ordering on L3 updates. Record with seq_num N should be applied to update the state of the book before doing the same with N+1. |
| add_order_id | *order_id* is unique for *side*. Can have same *order_id* for both BUY or SELL, treat them as 2 orders. |
| add_side | BUY or SELL |
| add_price | Price of this order |
| add_qty | Quantity of this order |
| update_order_id | *order_id* and *side* are used together to identify the original order it's trying to update |
| update_side | |
| update_price | |
| update_qty | |
| delete_order_id | *order_id* and *side* are used together to identify the order it's trying to delete |
| delete_side | |
| trade_order_id | *order_id* and *side* are used in conjunction to identify the (resting) order on the book this trade applies to (the order on the other side is irrelevant to the purpose of updating the L3 book) |
| trade_side | Used in conjunction with *order_id* to identify the (resting) order this trade applies to |
| trade_qty | Determines quantity to subtract from the (resting) order<br>*trade_qty* should be subtracted from the existing quantity of the (resting) order. If there is no quantity remaining after subtraction, the (resting) order should be considered removed from the L3 book |
| trade_price | |
| time | Timestamp in UTC |

# Explanation on L1 data

| Column | Description |
|---|---|
| time | Timestamp of L3 market data which triggers L1 market data update (in UTC) |
| bid_price | Best bid price |
| ask_price | Best ask price |
| bid_size | Total quantity for the best bid price |
| ask_size | Total quantity for the best ask price |
| seq_num | Sequence number of L3 market data which triggers L1 market data update |