Liquidity Lock Technical Problem Set

Suppose you are given a list of *orders* in a trading system, such as shown below:

Party	Direction	Product	Quantity	Price
A	Buy	Apple	10	98
B	Sell	Orange	15	100
C	Sell	Banana	5	102
B	Buy	Banana	8	105
E	Buy	Orange	10	103

The trading system will output a list of *trades* between party pairs e.g.:

Party	Counterparty	Direction	Product	Quantity	Rate
B	E	Sell	Orange	10	102

We define profit and loss (P&L) for a given trade to be

$$P\&L_{buyer} = Quantity \cdot (Price_{buyer} - Rate)$$
$$P\&L_{seller} = Quantity \cdot (Rate - Price_{seller})$$

For example, in the trade above party B is *selling* and their P&L is

10 * (102 - 100) = +20 P&L

and party E is *buying* and their P&L is

10 * (103 - 102) = +10 P&L

Question One

a) Describe an algorithm using pseudocode which matches as many buyers to sellers by creating trades between party pairs. Each trade must be executed at the mid-market rate, which is defined as the quantity-weighted average of input prices within a given product. For example, the quantity-weighted average of oranges in the above example is: $\frac{15 \cdot 100 + 10 \cdot 103}{15 + 10} = 101.2.$

b) Implement the above algorithm in a python script called **mid.py** which takes in orders in CSV format and outputs a set of trades in CSV format. List any assumptions you make at the beginning of the script.

c) Run your script on the sample order set (see Appendix) outputting a file called *mid_results.csv*

d) How much total volume was matched? How much P&L did each party make?

Question Two

Suppose that each party can now enter limits specifying how much net fruit they are able to buy and sell. For example, the table below gives a limit on party A's net amount of fruit they can buy/sell in the system.

Party	Direction	Net Limit
A	Sell	4
A	Buy	2

The following scenarios illustrate how the limits work:

- Suppose A buys 3 apples. This would violate the net buy limit of 2.
- Suppose A sells 5 bananas. This would violate the sell limit of 4.
- Suppose A buys 3 apples and sells 5 bananas. This would result in a net amount of selling 2 fruit, satisfying both sell and buy limits.

a) Describe an algorithm using pseudocode which, in addition to matching as many buyers to sellers at the mid-market rate, adheres to a set of total buy/sell limits entered into the system.

b) Implement the above algorithm in a python script called **limits.py** which takes in orders and limits in CSV format and outputs a set of trades in CSV format. List any assumptions you make at the beginning of the script.

c) Run your script on the sample order set and limit set (see Appendix) outputting a filled called *limits_results.csv*

d) How much total volume was matched? How much P&L did each party make? Which if any of the limits were hit?

Bonus Question

a) In a few sentences, how would you test each algorithm for correctness?

b) Write tests for each algorithm in a script called **tests.py**.

Appendix

Sample Order Set

Party	Product	Direction	Quantity	Price
A	Apple	Sell	16	96
A	Banana	Buy	1	104
A	Cherry	Sell	15	99
A	Orange	Buy	19	102
B	Apple	Sell	11	102
B	Banana	Buy	2	101
B	Cherry	Buy	9	99
B	Orange	Buy	10	105
C	Apple	Buy	20	101
C	Banana	Sell	11	104
C	Cherry	Sell	4	99
C	Orange	Buy	3	100
D	Apple	Buy	19	96
D	Banana	Sell	9	100
D	Cherry	Buy	1	97
D	Orange	Sell	12	99
E	Apple	Buy	13	105
E	Banana	Buy	9	101
E	Cherry	Sell	11	105
E	Orange	Buy	7	96

Sample Limit Set

Party	**Direction**	**Net Limit**
A	Sell	10

A	Buy	10
B	Sell	10
B	Buy	10
C	Sell	10
C	Buy	10
D	Sell	10
D	Buy	10
E	Sell	10
E	Buy	10