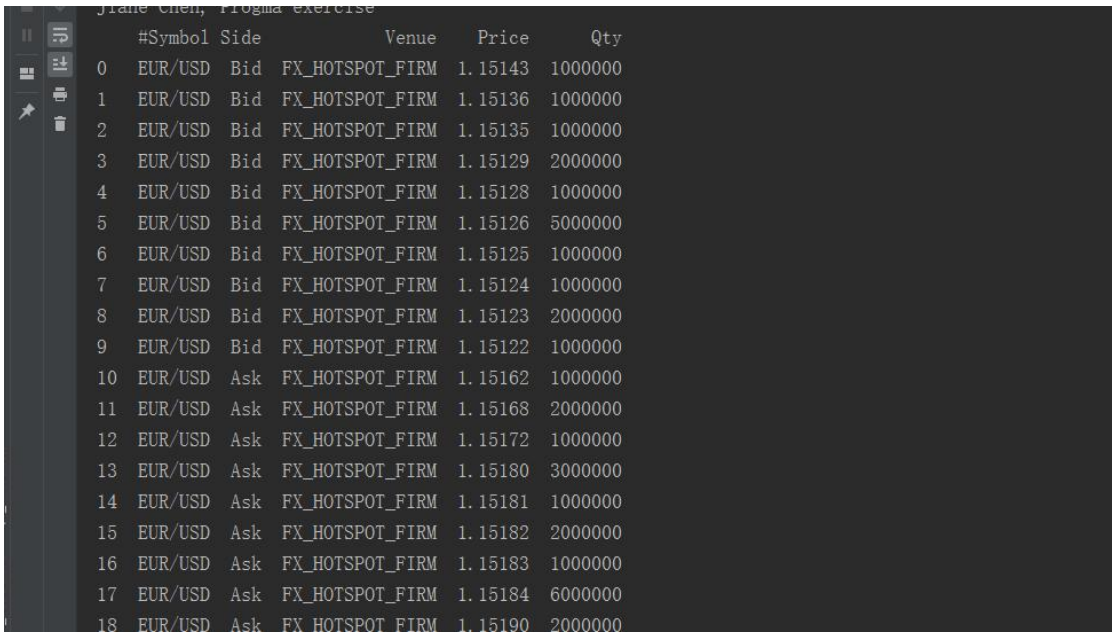
Thank you for reading my task document!

1. I put all the necessary details and output results in this PDF file. I also put all the codes at the end.
2. I used the python 3.7 and PyCharm in this case.
3. I also wrote the comments in the py file.

1. Read Data

Firstly, I transferred the data to a csv file which based on the UTF-8 and transformed file into Dataframe format so that I can process data in the pycharm environment.



```
Jiahe Chen, Trogma exercise
   #Symbol Side           Venue    Price      Qty
0   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15143  1000000
1   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15136  1000000
2   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15135  1000000
3   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15129  2000000
4   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15128  1000000
5   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15126  5000000
6   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15125  1000000
7   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15124  1000000
8   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15123  2000000
9   EUR/USD  Bid  FX_HOTSPOT_FIRM  1.15122  1000000
10  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15162  1000000
11  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15168  2000000
12  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15172  1000000
13  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15180  3000000
14  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15181  1000000
15  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15182  2000000
16  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15183  1000000
17  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15184  6000000
18  EUR/USD  Ask  FX_HOTSPOT_FIRM  1.15190  2000000
```

2.Process data

In this case, there are no unnecessary data and the data size is not big so I can view it clearly. Otherwise, I will filter the data

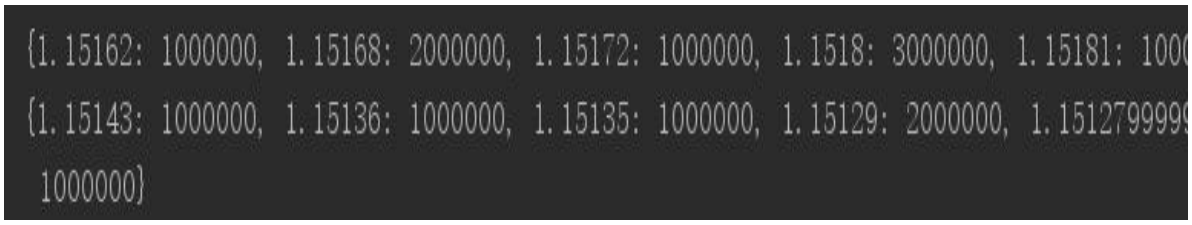
and refine the key information first.

I divided the data frame into two parts.

	#Symbol	Side	Venue	Price	Qty
10	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15162	1000000
11	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15168	2000000
12	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15172	1000000
13	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15180	3000000
14	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15181	1000000
15	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15182	2000000
16	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15183	1000000
17	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15184	6000000
18	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15190	2000000
19	EUR/USD	Ask	FX_HOTSPOT_FIRM	1.15203	1000000

	#Symbol	Side	Venue	Price	Qty
0	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15143	1000000
1	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15136	1000000
2	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15135	1000000
3	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15129	2000000
4	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15128	1000000
5	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15126	5000000
6	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15125	1000000
7	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15124	1000000
8	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15123	2000000
9	EUR/USD	Bid	FX_HOTSPOT_FIRM	1.15122	1000000

Then transferred the dataframe to lists and dictionaries and make two dictionaries for the prices and quantity of two sides(ask, bid). From the results we can see that all the prices have been sorted already, otherwise I will sort the data first.



2. Simulated transaction

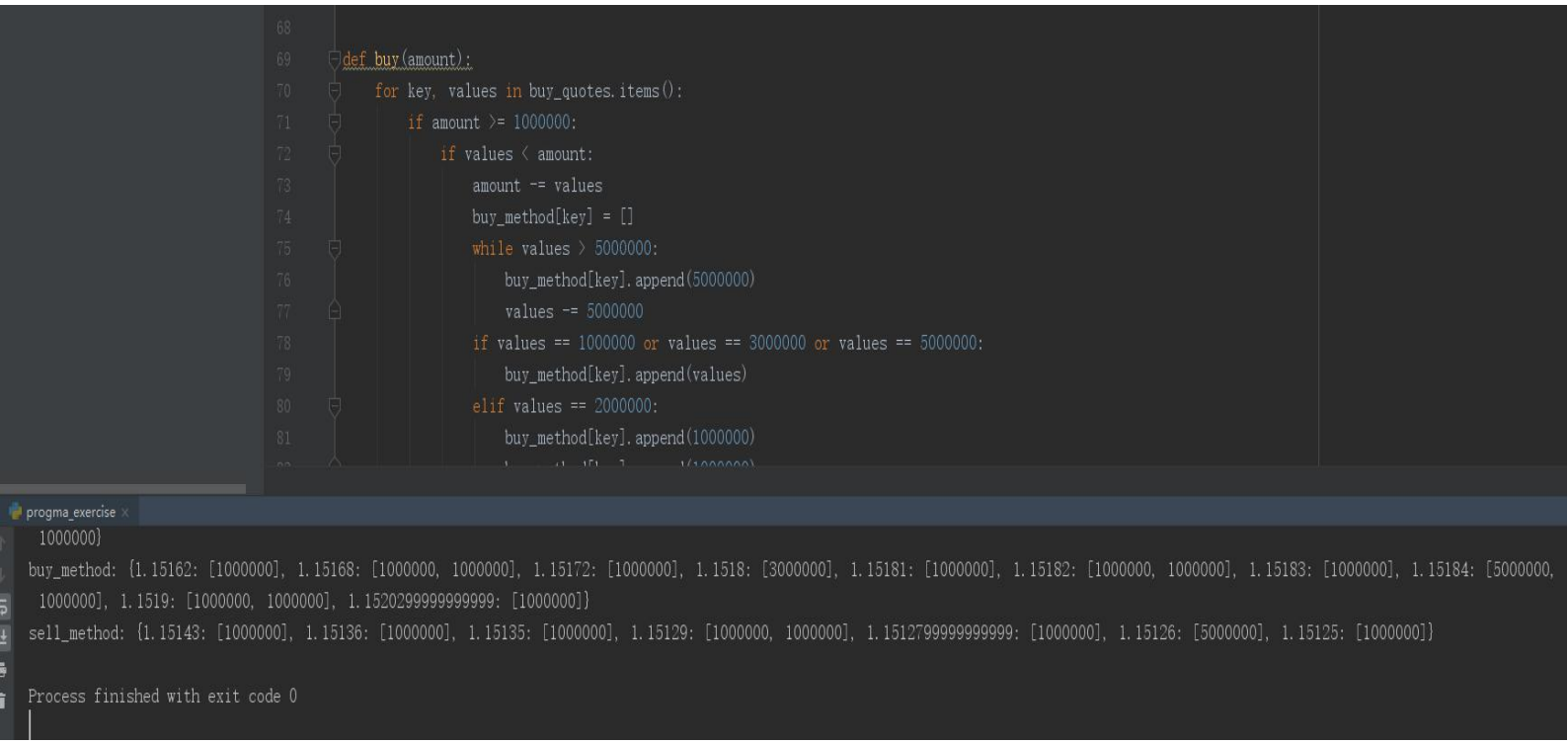
I wrote two functions to calculate the best method to buy and sell

1.As the requested, I used the 20 million dollars as the executed quantity for buying and 12 million dollars as the executed quantity for selling. And all the transaction will be in increments of 1, 3, and 5 million dollars.

2.I used two dictionaries to represent the results. The results show which quotes we should trade to fully execute the buy and sell trades against the best average price. I also put the quantity of every trade in the dictionary.

3. Since we are only allowed to take these quotes in increments of 1, 3, and 5 million dollars, some quotes have been taken more than once. For example, the number 17 (Ask FX_HOTSPOT_FIRM 1.15184 6000000) has been taken in increments of 1 and 5 million dollars. So this transaction will be shown as 1.15184: [5000000, 1000000].

Here is the output which shows the best method to buy and sell. It shows which price we should trade and how many quantity should we trade.



```
def buy(amount):
    for key, values in buy_quotes.items():
        if amount >= 1000000:
            if values < amount:
                amount -= values
                buy_method[key] = []
                while values > 5000000:
                    buy_method[key].append(5000000)
                    values -= 5000000
                if values == 1000000 or values == 3000000 or values == 5000000:
                    buy_method[key].append(values)
                elif values == 2000000:
                    buy_method[key].append(1000000)
```

```
progma_exercise
 1000000}
buy_method: {1.15162: [1000000], 1.15168: [1000000, 1000000], 1.15172: [1000000], 1.1518: [3000000], 1.15181: [1000000], 1.15182: [1000000, 1000000], 1.15183: [1000000], 1.15184: [5000000,
 1000000], 1.1519: [1000000, 1000000], 1.1520299999999999: [1000000]}
sell_method: {1.15143: [1000000], 1.15136: [1000000], 1.15135: [1000000], 1.15129: [1000000, 1000000], 1.1512799999999999: [1000000], 1.15126: [5000000], 1.15125: [1000000]}

Process finished with exit code 0
```

Thank you!

Jiahe chen.

2019/3/18