

Software Developer Technical Challenge

Problem Description

Your challenge is to write a trading algorithm that, given historical data, attempts to determine the best trades that could have been made to maximise profits. By looking at the data provided, your algorithm will aim to select the best times to enter the market and the best times to exit the market, in order to maximise the overall return. There are however some conditions that make this problem more complex. The following requirements must be observed:

1. A trade must be open for a minimum of 30 mins and closed before reaching 60mins.
2. You may only have 1 trade active at a time (eg, if you close at time 1:36 you can then open at 1:37).
3. You can only buy into the market, that is you can only make a profit from buying low and selling high.

The algorithm is only investing 1 'unit' in each trade. If your algorithm enters the market at 1.1 and exits at 1.2 then the algorithm has made 0.1 units of profit. Summing the return over all trades gives the algorithm's total return. Conditions 1 and 2 cause a situation where entering the market will commit you for a minimum time and other trades that overlap cannot be executed. This means each trade may come with tradeoffs.

Your task is to generate an application with your trading algorithm at the core. You will be provided data for a single 'stock' and must make trading decisions to maximise profit. All open trades must be closed before the end of the data.

Your program will:

1. Read in the data
2. Determine when to open a position
3. Determine when to close the position
4. Present trades and profit made.

Your trading algorithm is NOT required to produce the optimal result however you should be aware of any limitations your approach does have.

There are many ways to solve this problem. You are required to implement at least one profitable algorithm. If you like, you may implement multiple strategies and compare the results.

Data

The data is a CSV file with 2 columns, time and price. Time is the minute for the data point, for example, time=94 is 01hr 34mins. An example of what the CSV file looks like is (full data in external file provided):

```
Time,Price
0,1.2546
1,1.2567
2,1.2577
3,1.2579
4,1.2593
5,1.2668
6,1.2695
7,1.2689
8,1.2679
9,1.2679
10,1.2678
```

The following sample output was generated on a data set with 240 data points (Note: your output does **NOT** need to match this exact formatting):

Trades are:

```
Open at 0 (1.2546), close 58 (1.2796) for profit 0.025
Open at 64 (1.2635), close 94 (1.2845) for profit 0.021
Open at 101 (1.275), close 138 (1.2929) for profit 0.0179
Open at 139 (1.2846), close 169 (1.3016) for profit 0.017
Open at 178 (1.2943), close 228 (1.3728) for profit 0.0785
Total profit 0.1594
```

Expectations on Implementation

Please aim to spend no more than a few hours on this exercise.

Depending on the role you're applying for, you will be asked to implement your solution in either **Python** or **JavaScript**.

We respect your personal time and do not intend that this exercise uses more of it than necessary. We do not necessarily intend for candidates to produce production quality code. Nevertheless, we want you to demonstrate not just the core skills in the code, but also communicate your understanding of what it would take to produce a production tool. You can do this via placeholders, comments or other documentation as well as direct implementation.

Submission

To submit please either:

- Email us a zip file containing your solution.
- Share a GitHub repository containing your solution.

Please ensure your solution is accompanied by a README file containing clear instructions on how to run your solution and install any relevant dependencies.

Good luck! We look forward to seeing what you come up with.