Luiss
Libera Università Internazionale degli Studi Sociali Guido Carli

Algorithms A.Y. 2022/2023

Software Project – A Cryptocurrency Explorer and Market Analyzer

Irene Finocchi, Flavio Giorgi, Bardh Prenkaj
finocchi@luiss.it, fgiorgi@luiss.it, bprenkaj@luiss.it

07 February 2023

courtesy of: Andrea Coletta

LUISS

Dipartimento di Impresa e Management

Group Project Work

The project requires to:

- read a cryptocurrency dataset
- implement *efficient* algorithmic solutions to different problems

We release three different parts:

1. February 7 (Today) – due to February 28 (not mandatory)
2. February 28 – due to March 28 (not mandatory)
3. March 28 – due May 14 (mandatory)

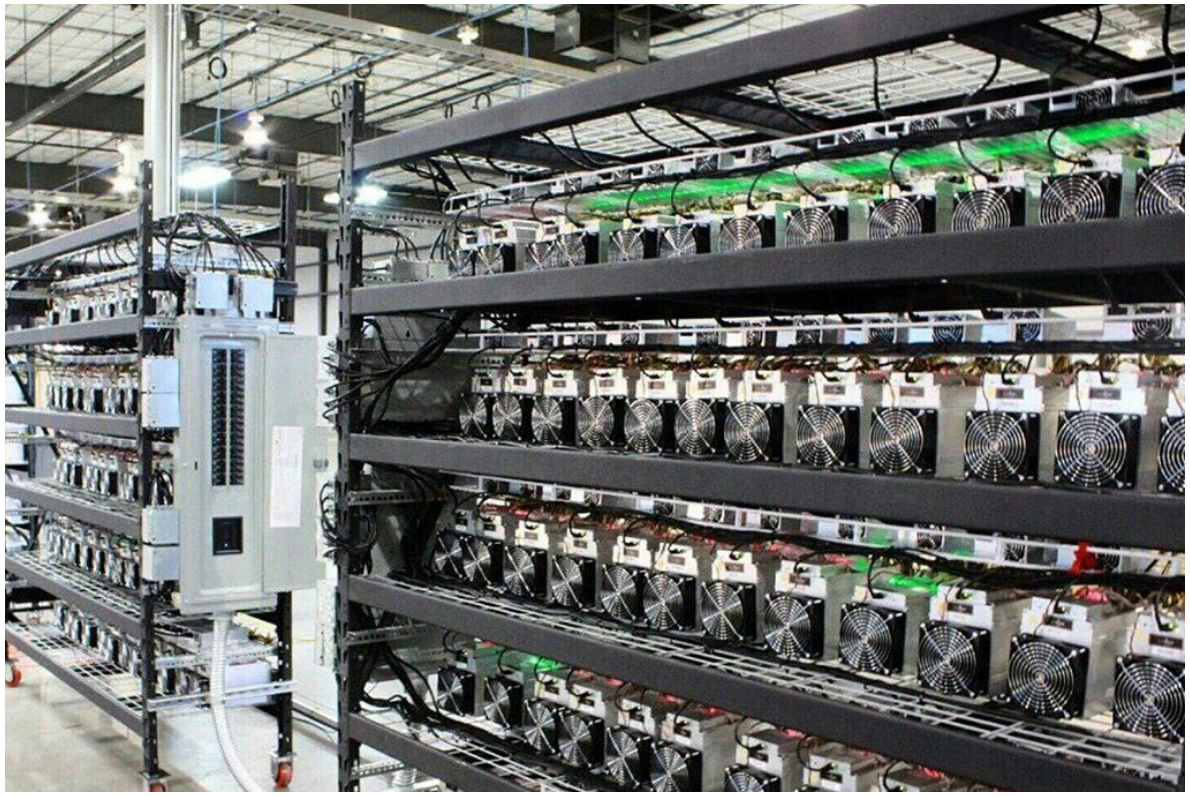
Group Project Work: A brief Introduction to the Crypto Market

A decentralized digital marketplace for buying and selling cryptocurrencies

Cryptocurrency prices are determined by supply and demand

Open 24/7, with trades made through online platforms and exchanges

Still in early stages of development and considered highly speculative and volatile



LUISS

Group Project Work: A brief Introduction to Crypto Market

LUISS

Group Project Work: A brief Introduction to the Crypto Market

The line represents the price of the cryptocurrency.

The crypto prices are determined in the marketplace, where seller supply meets buyer demand.

And the price of a crypto changes over the time according to the market, company performance, and users' actions (buy and sell).

The Figure shows an example of the price of Algorand (ALGO) listed for the entire day of 02/02/2023.

LUISS

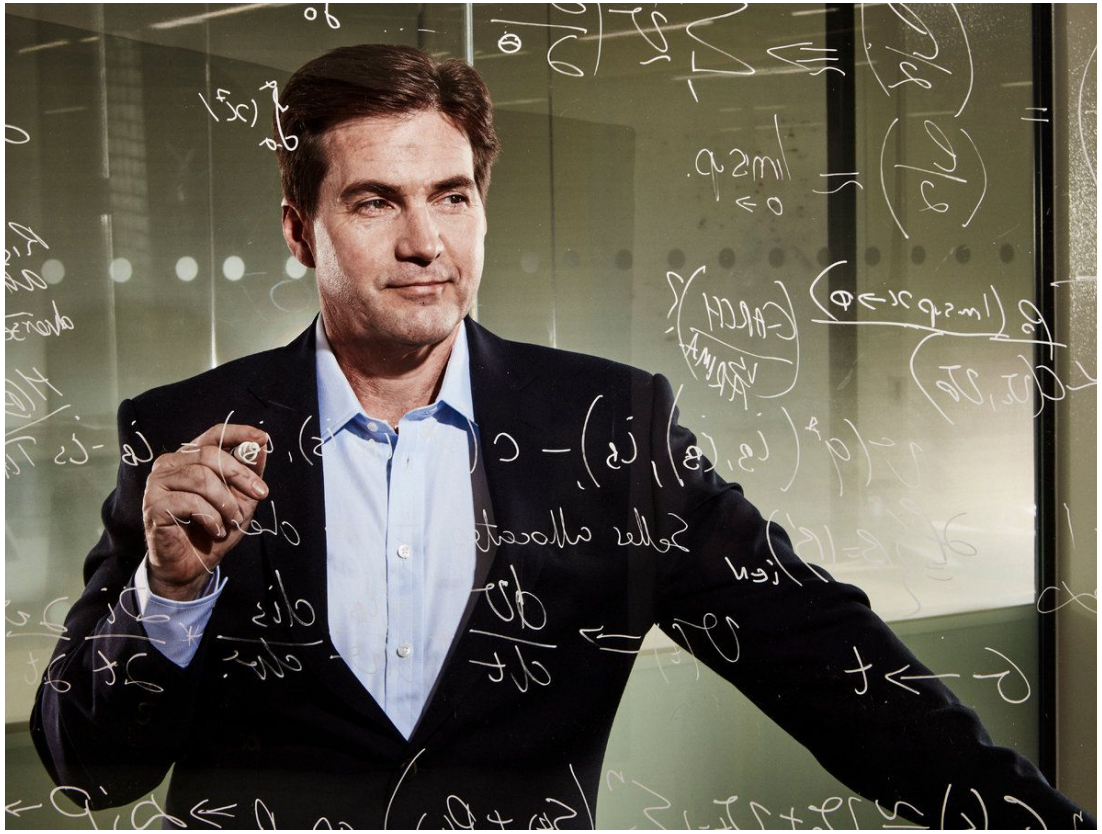
Group Project Work: A brief Introduction to the Crypto Market

In the most recent years, stock trading and decentralized trading has transitioned to fully electronic exchanges.

Therefore most of existing investment banks focus and aim at developing very efficient algorithms and fast solutions to trade stocks, and especially cryptos.

Question: do we really need fast and efficient algorithms for crypto market data?

Next question: who is the guy in the photo?



LUISS

A brief Introduction to the Crypto Market : Why efficient algorithms are really important?

Around 10 years ago (before 2009), a group of traders from Chicago traded stocks to New York Exchange (NYSE), using powerful computers and the existing internet connection (Figure 1).

They had fast algorithms, but they also needed around **17 milliseconds** to connect from Chicago to New York.

Figure 1 – Internet Connection between New York and Chicago

Therefore they decided to build a new optic-fiber line to reduce this time, from **17 to 13 milliseconds.**

It cost **300 Millions** of dollars and **1300 km** of new cable for just **4 milliseconds**.

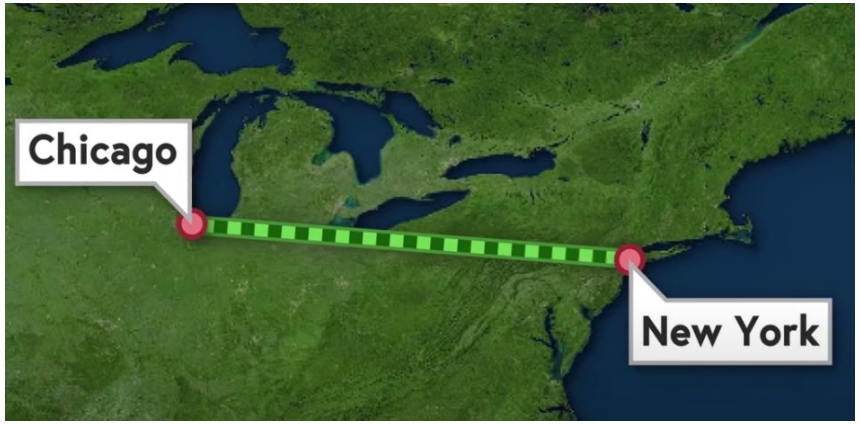


Figure 2 – **New FAST** Internet Connection between New York and Chicago

LUISS

Software Project: Input Data

How to store financial data in a simple way?

Software Project: Input Data

How to store financial data in a simple way?

You are given as input a .txt file containing a list of stocks and additional details. Each line has:

crypto_name, day, price, volume

The values represent the price and volume for the crypto_name (e.g., ALGO) in that day.

Crypto	Day	Price	Volume
Gala	458	45	5559100
1inch	507	288	1938100
Etherium	464	75	3553000
Bitcoin	723	65	18966800
Gala	397	97	1314100
Algorand	588	1290	0
Algorand	581	1290	0
Etherium	727	504	0
Tether	643	1398	0

Software Project: Input Data

You are given four datasets*: dataset_small.txt, dataset_medium.txt, dataset_large.txt, dataset_full.txt*:

Filename	Size	Rows	Cryptos	Average monitoring days per crypto
dataset_small.txt	901KB	~24k	98	247
dataset_medium.txt	~1.8MB	~48k	98	492
dataset_large.txt	~2.6MB	~69k	98	707
dataset_full.txt	~4.60MB	~120K	98	1232

In the first release we suggest to use only the “dataset_small.txt”.

Group Project Work : Task 1

In the first project release we require to design and implement :

1) **A Python function that reads the TXT file**, extracting the relevant information which must be then stored in a suitable data structure.

2) **A python function to answer queries of the form**: Which are the *min-price, average-price and max-price* values for the crypto **C** in the time interval **[a,b]**?

Group Project Work : Task 1

Example:

1) Which are the min, average, max values for **Gala** price?

2) Which are the min, average, max values for **Algo** price?

Your code should answer:

1) 191, 196.14, 200

2) 268 , 279.43, 292

Stock	Day	Price	Volume
Gala	371	200	0
Algo	369	275	13038300
Algo	370	273	0
Algo	371	273	0
Gala	369	197	18526600
Gala	370	200	0
Gala	365	191	27862000
Algo	365	289	8110400
Algo	366	286	5478900
Algo	367	292	7929900
Algo	368	268	23720700
Gala	366	194	22765700
Gala	367	195	23271800
Gala	368	196	19114300

Group Project Work : Implementation

We are providing you a skeleton of the code. You should modify the code we provide, adding the missing parts.

In particular, you have to implement **“group0.py”**. You can use this file to also add your helper functions.

LUISS

You can download the code from Luiss-learn or Github:
https://github.com/flaat/Algorithms-2022-2023-Project

Group Project Work : Implementation

The python file **“group0.py”:**

The function read_file is called **once** to load the dataset: it can be used to prepare and read the input file (e.g., *“data/dataset_small.txt”*).

```
def read_file(file_path: str) -> any:
    """
    This function reads the dataset containg all the information about the
    cryptocurrecies. The information are stored in a .txt file.

    Parameters:
    :file_path: The current path where the file you want to read is located

    @return: A data structure contining the information of the crypto
    """

    # TODO: Implement here your solution
    return None
```

LUISS

Group Project Work : Implementation

The python file **“group0.py”:**

The function **crypto_stats** implements your query algorithm!

It receives as input the stock name (e.g., *”Gala”*) and a time interval in days (e.g., [15,18]), it outputs the observed min, average, max of the price in the time interval. The order of the output is important!

```python
def crypto_stats(data, crypto_name: str, interval: Tuple[int, int]) -> Tuple[float, float, float]:
    """
    This function calculates the minimum, average, and maximum price values of a crypto
    whose name is passed in input within a specific period of time [a,b] passed      You, 2 weeks
    in input. Notice that [a,b] can be an interval that might exceed the actual monitoring
    time of the crypto given in input.

    If any error occurs, return the default value (0.0, 0.0, 0.0)

    Parameters:
    :data: The data structure used to calculate the statistics
    :crypto_name: The name of the cryptocurrency to calculate statistics for
    :interval: The time interval consisting of a tuple of two values (a,b)

    @return: A tuple that contains the minimum, average, and maximum price values
    """
    # TODO: Implement here your solution

    return None
```

LUISS

Group Project Work : Execution

Once implemented, you can execute your project using the following command:

“python3 grader.py”

This will test your code.

You receive a **text** and **visual** feedback.

```
PS C:\Users\39348\OneDrive\Documents\GitHub\Algorithms-2022-2023-Project> python .\grader.py
reading_file --- Elapsed time: 0.071860 seconds for dataset full
reading_file --- Elapsed time: 0.072803 seconds for dataset large
reading_file --- Elapsed time: 0.049045 seconds for dataset medium
reading_file --- Elapsed time: 0.016000 seconds for dataset small
Test failed.
You correctly returned the data up until index=0.
2.371590995788574 is not equal to 1.374856948852539
Test failed.
You correctly returned the data up until index=0.
0.2862812998890877 is not equal to 0.22393299639225
Test failed.
You correctly returned the data up until index=0.
11.486145973205566 is not equal to 10.727279663085938
Test passed on dataset_small.
Test failed.
You correctly returned the data up until index=0.
0.9987026651700338 is not equal to 0.9942460060119628
Test failed.
You correctly returned the data up until index=0.
4.080670619570969 is not equal to 0.8875120282173157
Test failed.
You correctly returned the data up until index=0.
1.2066967520448897 is not equal to 1.0118930339813232
```

Group Project Work

Report:

- For the first two parts, you can write a simple presentation about your implementation (about 2 slides). NOT MANDATORY, but HIGHLY SUGGESTED!

- The final release is MANDATORY TO PASS the Project, and you must provide a presentation with at most 8 slides.

Describe your algorithmic idea, main implementation details, and experiments. You should try to analyze the asymptotic cost of your implementation.

Group Project Work

Score:

- The project contributes up to 8 points (added to the theory score).
- If you miss the May deadline, the maximum grade is lowered: you can achieve max 6 points (if you deliver the project by the second exam session), max 5 points (third session), max 4 points (fourth - and last - session)
- For **top projects**, we might consider assigning an extra **1-point bonus**.

You should work in groups of 3 students.

LUISS

More details on Luiss-learn and Project guidelines pdf.

Thank you!

LUISS