Algorithms A.Y. 2021/2022
BSc in Management and Computer Science
Luiss Guido Carli

Software Project – A Stock Market Data Analyzer

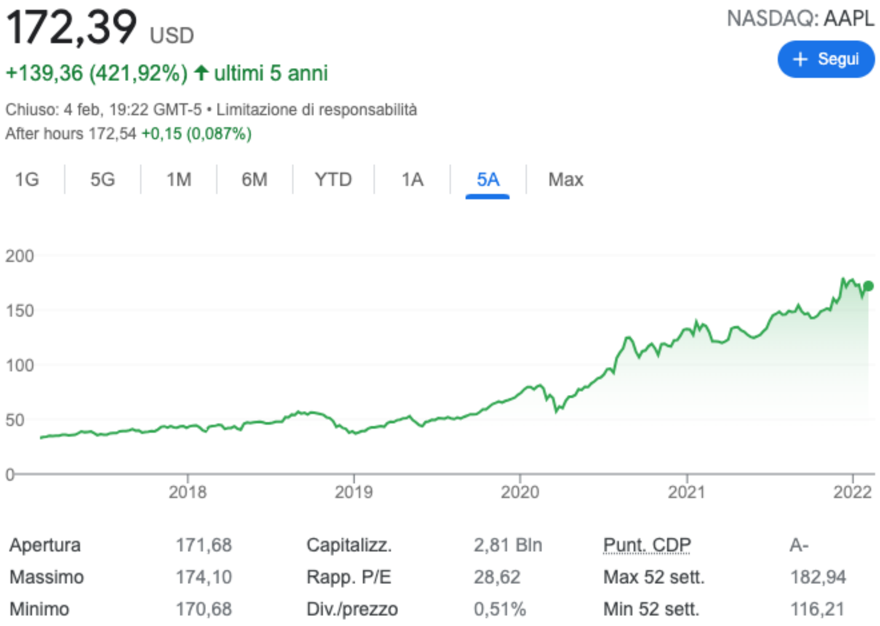
Final Release

Andrea Coletta, Irene Finocchi
acoletta@luiss.it, finocchi@luiss.it

1. A Brief Introduction to Stock Market

A stock market is a place where buyers and sellers meet to exchange equity shares of public corporations. An equity (also known as stock) is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own. Units of stock are called "shares."

Stock prices are determined in the marketplace, where seller supply meets buyer demand. For example, there may be three buyers who have placed orders for buying Apple shares at $172.39, and there may be four sellers who are willing to sell Apple shares at $172.39. The two offers match, and the transaction defines the new price of Apple at 172.39$.



172,39 USD

NASDAQ: AAPL

+139,36 (421,92%) ↑ ultimi 5 anni

Chiuso: 4 feb, 19:22 GMT-5 • Limitazione di responsabilità

After hours 172,54 +0,15 (0,087%)

Apertura	171,68	Capitalizz.	2,81 Bln	Punt. CDP	A-
Massimo	174,10	Rapp. P/E	28,62	Max 52 sett.	182,94
Minimo	170,68	Div./prezzo	0,51%	Min 52 sett.	116,21

Figure 1 – An example of Stock Data

Thus, the price of a stock changes over the time according to the market, company performance, and users' actions (buy and sell). Figure 1 shows an example of the price of Apple (AAPL) listed at NASDAQ market (US) from 2017 up to February 2022.

A stock market usually contains several stocks (i.e., companies). For example, the biggest market in US, called NASDAQ, lists around 3500 stocks including Apple (AAPL), Meta (FB), Microsoft (MSFT), Google (GOOGL), and Netflix (NFLX).

When investing into the financial market we usually create a ***Portfolio.*** A portfolio is a collection (i.e., a set) of financial investments like stocks, bonds, commodities, or exchange traded funds (ETFs). A portfolio may contain a wide range of different stocks, and you may choose to hold and manage your portfolio yourself, or you may allow a money manager, financial advisor, or another finance professional to manage your portfolio.

One of the key concepts in portfolio management is the wisdom of diversification—which simply means not to put all your eggs in one basket. Diversification tries to reduce risk by allocating investments among various financial instruments, industries, and other categories. It aims to maximize returns by investing in different stocks that would each react differently to the same event (uncorrelated stocks)!!

The goal of this project is to study correlation among stocks, and answer a fundamental question: If I have a stock X in my portfolio, which stocks should I avoid/sell to reduce risk? Therefore, we manage our portfolio by studying correlations, which represent the degree of relationship between the price movements of different assets.

2. **Project Data**

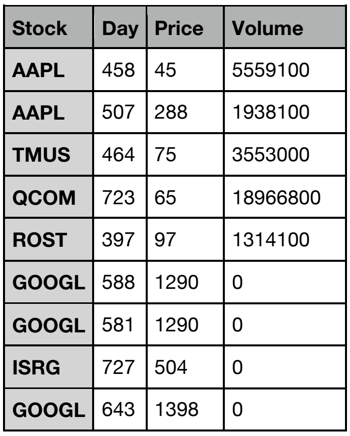
The project requires reading a financial dataset and implementing efficient algorithmic solutions to different problems in Python, also experimenting with a few real datasets of different, increasing sizes.
In detail, you will analyze a set of US stocks, and compute several metrics.

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

stock_name, day, price, volume

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

For simplicity, *days*, *prices*, and *volumes* are all integer values, while *Stock* should be read as type string. The volume defines the amount of shares traded on that day. Volume 0 means that no transactions took place on a given day. You cannot assume any ordering in the provided data.



Stock	Day	Price	Volume
AAPL	458	45	5559100
AAPL	507	288	1938100
TMUS	464	75	3553000
QCOM	723	65	18966800
ROST	397	97	1314100
GOOGL	588	1290	0
GOOGL	581	1290	0
ISRG	727	504	0
GOOGL	643	1398	0

Figure 2 – Input Stock Data

You are given four datasets: small_dataset.txt, medium_dataset.txt, large_dataset.txt, huge_dataset.txt:

- The small dataset has about 50 stocks with 1 year of data.
- The small dataset has about 500 stocks with 1 year of data.
- The large dataset has about 5000 stocks with 1 year of data.
- The full dataset has about 6000 stocks and 2 years of data.

We will run your query algorithm on the different datasets: the larger the dataset you can handle (correctly), the better the evaluation of your project! We will also consider the running time of your code on the datasets that you can solve.

3. Risk management and Portfolio Diversification

As we mentioned, diversification is a way for investors to reduce risk. The asset values within a well-diversified portfolio do not move up and down in perfect synchrony. Instead, when some assets' values move up, others tend to move down, evening out large, portfolio-wide fluctuations and thus reducing risk.

Therefore, in our portfolio if we buy PEPSI probably, we don't want to buy CocaCola. They are both automotive and high correlated!

Stock correlation.

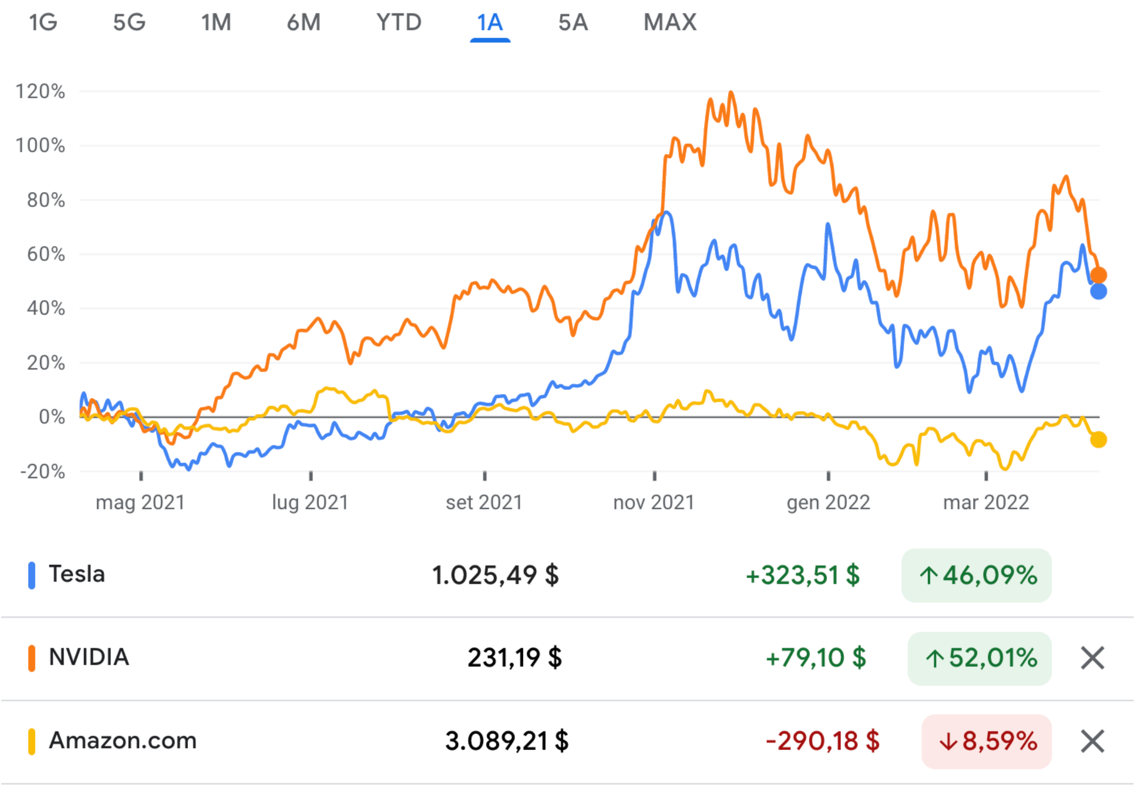
We now introduce a simple correlation metric for two stocks.
We first define the return $\underline{r}_s$ for a stock s over a period of N = {0, .. , n} days as follows:

$$r_s = \begin{cases} (x_n - x_0) \,/\, x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

which is calculated by subtracting the final value of the stock from its initial value, and then we divide this new number by the initial value.

This number measures how much a stock gains during the year. Some stocks increase their value ($\underline{r}_s > 0$) others lose value ($\underline{r}_s < 0$).

Intuitively, two stocks are correlated if they have a similar $\underline{r}_s$ over the entire period. This means that they move together in the market:



Tesla	1.025,49 $	+323,51 $	↑46,09%	
NVIDIA	231,19 $	+79,10 $	↑52,01%	✕
Amazon.com	3.089,21 $	-290,18 $	↓8,59%	✕

In the above chart, we can see that both TESLA and NVIDIA are correlated and their return r_s is about 50%, while AMAZON is less correlated to them with a return r_s of -8.5%.

The general idea is that we do not want to keep both TESLA and NVIDIA in our portfolio. Even if they have outstanding performance, they are too similar! If the market falls, they probably fall as well. On the other hand, if we keep AMAZON and TESLA in our portfolio, we are probably less exposed to market crashes.

Therefore, we can define a distance score (similarity) between stock i and stock j as the absolute difference in their returns:

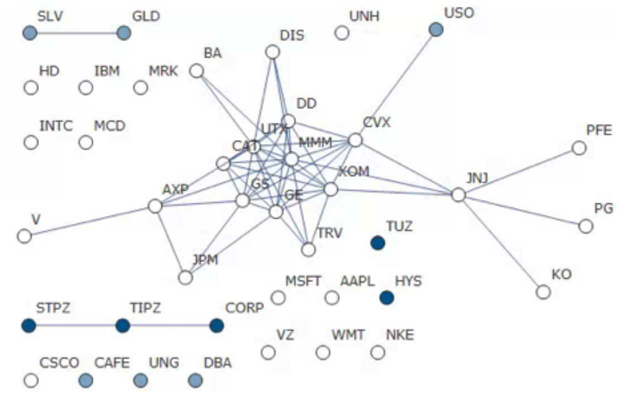
$$c_{ij} = \; | r_i - r_j |$$

According to the previous example, the score between NVIDIA and TESLA is |0.52 – 0.46| = 0.06 (very similar) while the score between AMAZON and NVIDIA is |-0.08 – 0.52| = 0.6 (they are much more distant).

Thus: the less the score the higher the correlation! For simplicity, we'll say that two stocks are correlated if their score c_{ij} is less than a threshold t (i.e., $c_{ij} < t$).

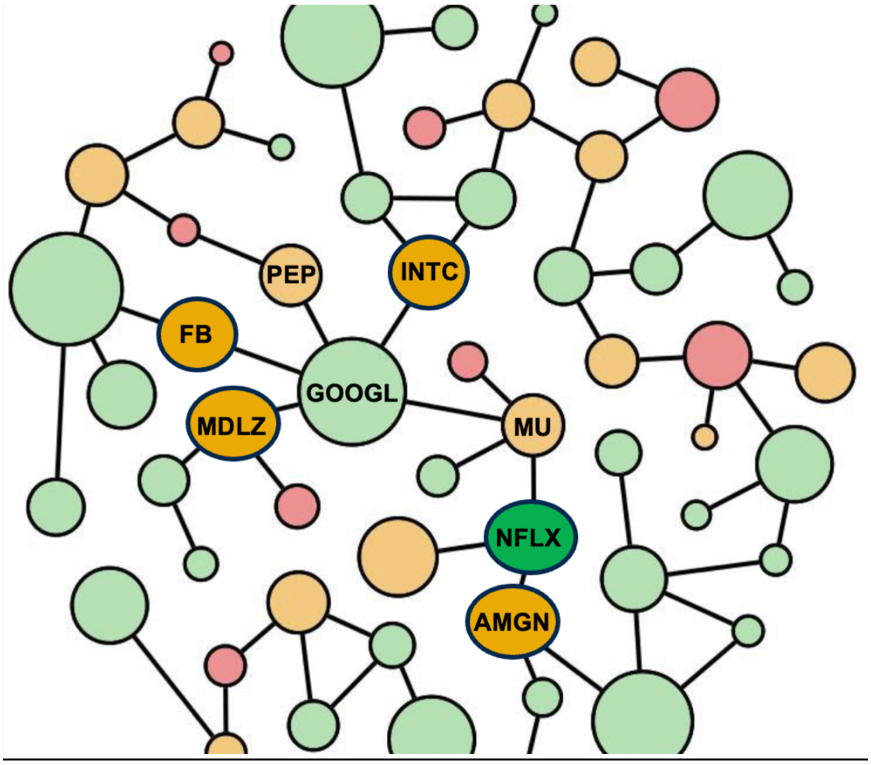
Correlation Graph:

To study correlation, we often use a graph that maps the interrelations between assets that are correlated at some specified threshold t (0.5 in this illustration).



Obviously, the choice of correlation threshold is somewhat arbitrary, and by changing it we can obtain different graphs. **This threshold would be an input for your function!**

A graph/network is a collection of nodes and arcs that connect those nodes. In the context of equities, each node represents a different stock and each edge represents an existing correlation between two stocks



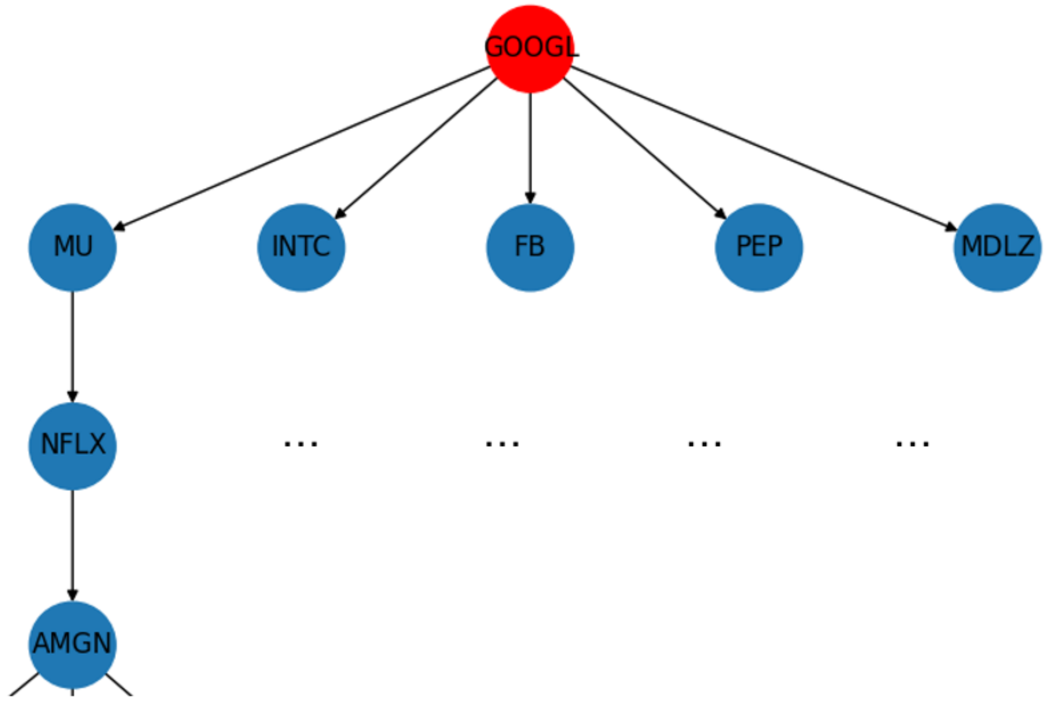
It is interesting to see how the graph is a very useful structure to understand the correlation between stocks!

In the chart above, we can easily see how two stocks are correlated! For example, GOOGL is directly correlated with MDLZ, FB, MU, INTC, and PEP. Nevertheless, MU is correlated also with NFLX!

Therefore, we have that:

- If a crash happens on FB is very likely to have a crash also on GOOGL
- If a crash happens on NFLX is very likely to have a crash on MU, and therefore the crash propagates on GOOGL as well!

How can we study these phenomena? We can build a Tree:



The tree has GOOGL as the root, and at level 1 we have the stocks that are directly correlated (MDLZ, FB, MU, INTC, and PEP.); at level 2 we have the stocks that are correlated through a single node (e.g., NFLX is correlated through MU); at level 3 the stocks that are correlated through 2 nodes (e.g., AMGN is correlated to NFLX, that is correlated to MU and finally to GOOGL).

In general, at level i we have the stocks correlated with the root through i-1 intermediate nodes!

Task:

You have to design and implement a Python code that reads the TXT file, extracts the relevant information, computes the correlation between all pairs of stocks, and stores them in a suitable data structure.

The data structure should make it possible to answer queries of the form:

Given the stock X, which are all the correlated stocks at level i?

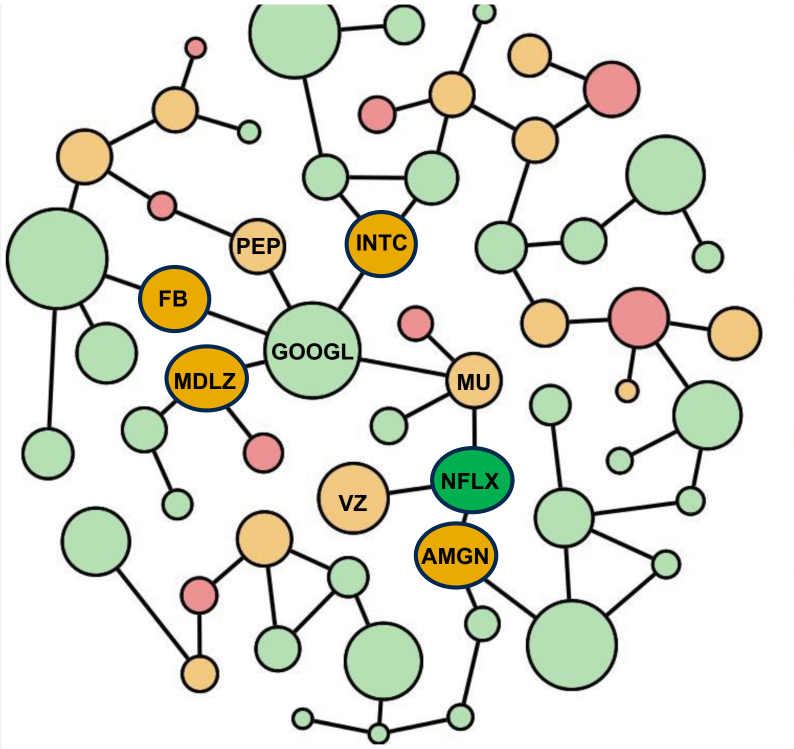
Notice that the query should return the list of stocks in alphabetical order.

Thus, given the input dataset, the goal of our project is to read the dataset, compute the correlation among each couple of stocks, and organize the stocks in a data structure to easily identify the levels and correlated stocks.

For simplicity we remind that the most important steps are:

a. Read the dataset
b. Compute the returns
c. Compute the correlation using the threshold t
d. Create a suitable data structure for the stock correlation (E.g., Graph)
e. Answer the query by visiting the graph.

Example:
If we consider the following graph of correlations computed with threshold 0.5:



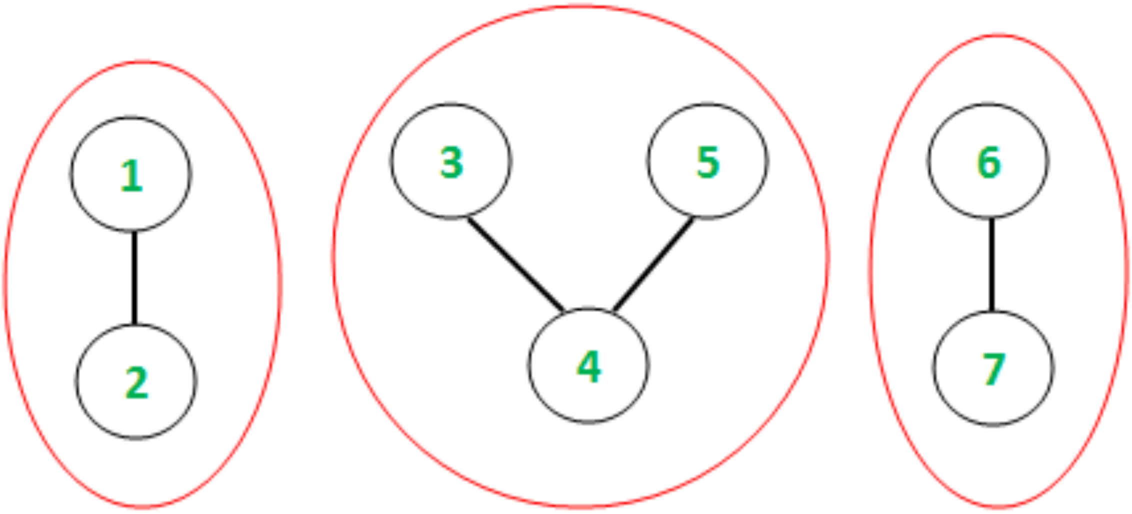
And if the query asks:

1) What stocks are correlated to GOOGL at level 1?
2) What stocks are correlated to MU at level 2?

And the answer should be:

1) ['FB', 'INTC', 'MDLZ', 'MU', 'PEP']
2) [‘AMGN’, 'FB', 'INTC', 'MDLZ', 'PEP', ‘VZ’]

4. Optional Part



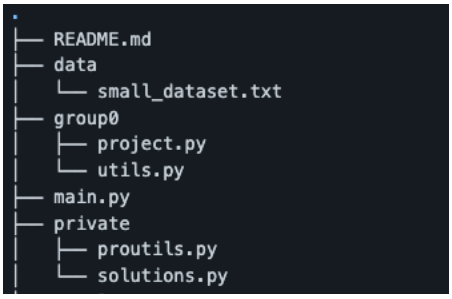
Looking at the correlation graph we often find groups of highly inter-correlated stocks that we can easily identify. Is crucial to investigate these clusters for several reasons, for example to build a portfolio we can select stocks belonging to different groups and thus avoid correlations and reduce risk.

In this optional part of the project, you are required to compute the number of groups (i.e., connected components) that are in the correlation graph.

5. Python implementation guidelines

You can download the code from Luiss-learn or github:
https://github.com/Andrea94c/Algorithms-2021-2022-Project

The project has the following structure:



```
.
├── README.md
├── data
│   └── small_dataset.txt
├── group0
│   ├── project.py
│   └── utils.py
├── main.py
├── private
│   ├── proutils.py
│   └── solutions.py
│
```

We are providing you a skeleton of the code. You can modify the code we provide, adding the missing parts (mostly inside group0/project.py).

In the project folder, you will find two python scripts to execute your project:
- a file main.py, which is used to run some simple tests for your project and check if your solution is correct
- a file times.py, which is used to evaluate the efficiency of your approach

The input datasets are inside the data folder (e.g., data/small_dataset.txt). You should not change the main file, except for the variable group_id (as specified below).
Change the name of folder group0 to match the group id that we will assign you after the registration on Luiss Learn, and also change the group_id value in main.py.

Implement your code in file groupid/project.py, which contains three functions (you can add more functions if needed, but you must at least implement these ones).

- Function ***prepare*** will be called once to load the dataset: it can be used to prepare and read the input file (e.g., data/small_dataset.txt), and store the relevant information in suitable global data structures of your choice. It takes as input the dataset filename and the threshold t to compute the correlation.

- Function ***query*** should implement your query algorithm, as described in the project pdf. It receives as input the stock name (e.g., AAPL) and a correlation level. It outputs the ordered list of correlated stocks. Thus, the order of the output is important! The list should be in alphabetical order!

- Function ***num_connected_components*** is optional (not mandatory to implement), and it should return the number of connected components in the correlation graph: a connected component is a set of stocks that are linked (correlated) to each other.

You can use additional files if needed, but all of them must be in the group folder. There is a file utils.py where, if you want, you can implement auxiliary algorithms and data structures.

You can use Python lists, dictionaries, and string functions, but no specific algorithm from any Python library! You can't directly use external libraries, if you need an external tool you must ask by email. If you need an algorithm (e.g., for searching, sorting, or selection) you must implement your own version from scratch. If in doubt, just ask.

When you run the main.py script you receive textual feedback about your solution (e.g., wrong, or correct, and expected results).
Moreover, at the end of the execution you receive feedback on how many tests you correctly completed:
E.g.,:
Final result: 18 / 18 correct solutions!

NOTE: the number of tests changes if you enable/disable the optional ***num_connected_components*** function. You can enable/disable it in the main.py by setting the variable OPTIONAL_TEST to True/False.

Please install the following python packages:

```
pip install pandas
pip install seaborn
pip intsall numpy
pip install matplotlib
```

6. Project Report

This final project IS MANDATORY to pass the Project. You must provide a presentation with at most 6 slides, describing your algorithmic idea, main implementation details, and experiments. You should try to analyze the asymptotic cost of your implementation.

You should send us (Coletta and Finocchi) an e-mail with a pdf file containing your presentation/slides and a zip file with your group folder. When you send us your project, please, use as subject "Algorithms 2022: group X", where X is your group ID assigned by the instructors.

More details on Luiss-learn.