

INITIAL THOUGHTS

- I need to keep track of processing time of each URI. So, I will need a private map attribute where **keys are URIs** and **values are vector of processing times**.
- I cannot override **simulateLatency** as that is a private method. I also cannot override **process** since it calls **simulateLatency**.
- I can only override **start()** and **finish()**. So, I will start a timer in **start()** and keep track of that as an attribute. In finish, I will get the current time and use subtraction to get the processing time and store that in my map attribute.
- To add elapsed time to the map, the finish method needs access to the URI. I can copy URI to a private attribute in **start()**. That way finish will have access to it.
- If I have a map containing URIs and timings, making mean and SD methods will be easy.
- To plot a histogram, Python's **Matplotlib** library is common. I need to figure out a way to use that via my C++ project.
- An easy way to conduct light weight unit tests is via **doctest.h**. I will just need to copy of the third-party **doctest.h** file to my project folder.

IMPLEMENTATION PROBLEMS

- I am using **Matplotlib wrapper matplotlibcpp.h** to plot the histogram
- But the hist function does not seem to let me plot normalized histograms
 - It enables drawing of regular histograms
- I then planned to use the bsar() function to make my normalized histogram
 - But the bar() function C++ wrapper does not seem to accept bar width as input. So, my final output looks more like a bar chart instead of a histogram.