Risk Analyst - Problem Solving

This section is a coding challenge, there are two parts of this problem, the first one is mandatory, the second one is a challenge problem.

In the first problem, you will be asked to implement a simple `CBArray` class with some requirements. The second problem is optional as an extra credit, you will be asked to extend `CBArray` class to `CBArrayEasyMin` with additional functionality.

Code templates and detailed requirements are provided in the implementation files in `src` folder: `cb_array.py` and `cb_array_easy_min.py`.

(Bonus points if you can use Git to track your changes.)

UnitTest

Unittests are also provided as a detailed guide of expected behavior.

1. Setup

To setup the correct environment to run unittests, follow these steps:

1. Make sure you have `Python 3` installed on your computer, if not, you can download it from here: https://www.python.org/downloads/. (we recommend Python 3.6)
2. After installation, go to Terminal on macOS or Linux, or PowerShell on Windows. Run `python --version` or `python3 --version`, make sure `Python 3.6.x` is displayed.
3. Note: if `python3 --version` gives you the correct Python version, for all following commands, replace `python` with `python3`, and `pip` with `pip3`.
4. Run `sudo pip install virtualenv --upgrade` to install `virtualenv` package, this allows us to easily install other packages we need to run our unittests.
5. Run `cd path/to/your/project/directory` to change your working directory to projct directory.
6. Run `python -m virtualenv venv` to initialize the virtual environment.
7. Run `chmod +x run_tests.sh` to make the script executable.

2. Running UnitTests

After setting up the environment, simply run `./run_tests.sh` to run all unittests against your implementation.

There are two testing files, one for `CBArray`, one for `CBArrayEasyMin`. Please try to get all unittests pass for `CBArray`, and if you decided to also work on `CBArrayEasyMin`, please try to pass all unittests in both files.

Note

There is no "correct" implementation, as long as your implementation passes corresponding unittests and satisfies all requirements, it is considered acceptable. If your code doesn't pass all

unittests, that's also fine, but we encourage you to try to pass as many as possible.

If we do decide to continue the interviewing process, please be prepared to explain your thought process and why you made certain implementation choices rather than other ones.

Thank you and good luck!