Technical Assessment

Prompt

Write a command line program that uses GitHub's REST API v3 to find the top N contributors across a GitHub account's open source repositories, based on the sum of their contributions to each repository.

To minimize its runtime, the program should make as many API requests as allowed concurrently, but to be a well-behaved API consumer it should obey a limit on the maximum number of concurrent requests.

The program should write each contributor's username and total contribution count to stdout, one per line.

Submission format

Submit the solution as a zipped or tarballed git repository, containing source code and a README (in whatever format you prefer).

Expectations

We know you're a busy person, so we hope that you won't spend more than a couple of hours working on this!

Please use what programming language is most comfortable for you. We're not looking for perfect, production-ready code, but we would like to see a submission with clear attention to detail and reasonable decisions and tradeoffs around performance, implementation complexity, testability, and readability.

Please treat the README as you might a pull request description, providing useful context for reviewers such as instructions for running the program, design decisions made, and tradeoffs considered.

Requirements

- The program should take 4 command line arguments:
    - **`--owner`**, the owner (i.e. user or org) whose repositories will be analyzed

    - **`--limit`**, the number of results to return
    - **`--concurrency`**, the maximum number of concurrent requests to make
    - **`--access-token`**, the
- Ensure that the program is a well-behaved consumer of the API by making no more than the specified number of HTTP requests
- Each contributor's username total contribution count (across all repositories) should be written to stdout, one per line, in the form **`<username>,<count>`**
- Pagination is required to gather the entire set of repositories for a given owner *and* to gather the entire set of contributors for a given repository
- Feel free to use 3rd party packages (e.g. in Python, using requests to make HTTP requests is fine)
- Do not use GitHub's newer GraphQL-based API
- Any error or status output should be written to stderr (not stdout)

Test parameters

Assume that the program will be tested with these parameters, along with a valid access token:

- **`--owner=awslabs`**
- **`--limit=10`**
- **`--concurrency=5`**

The AWS Labs organization was chosen because it has a large number of interesting open source repos.

Tips

- Create a "Personal Access Token" (instructions) to use for authentication (docs).
    - If you do not have a GitHub account and would rather not create one, please let us know and we will provide a valid access token to use!
- An account's repositories may be fetched from https://api.github.com/users/{owner}/repos and a repository's contributors may be fetched from https://api.github.com/repos/{owner}/{repo}/contributors.