[CS 11 25.1] Mock HOPE 2d – Project Oiler

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Submit solution
[CS 11 25.1] Mock HOPE 2

My submissions

✓ **Points:** 200 (partial)
Time limit: 6.0s
Memory limit: 2G

Problem type

Allowed languages
py3

Problem Statement

There is a site called *Project Oiler*, where people solve a bunch of math puzzles by programming!

In Project Oiler, there are q questions, numbered 1 to q. All questions have a fixed integral answer.

You are asked to maintain the leaderboard for Project Oiler, and as such, you are tasked to process two kinds of events efficiently.

- For the first kind of event, you are given a string u and two integers p and a. This means that the person with username u submits an answer of a to question number p.
- For the second kind of event, you are given an integer k. You must determine the k^{th} ranked person in the leaderboard.

The *score* of a user is the number of questions they have correctly solved.

Note that the leaderboard is sorted by (1) score (the larger the better), and (2) in case of ties, the timestamp of the latest submission (the earlier the better), regardless of whether the submission was correct or not.

Task Details

Your task is to implement a function named `maintain_leaderboard`, which should have the following *signature*:

```
def maintain_leaderboard(answers, events):
```

Copy

The above says that it has two arguments `answers` and `events`.

- `answers` is a length-q list of integers denoting the answers to the q questions.
- `events` is a list of e tuples. Each tuple is one of two kinds:
  - `('submit', (u, p, a))`, denoting that the person with username u submits an answer of a to question number p.
  - `('rank', k)`, which asks you to return the person with rank k on the leaderboard.

The function is expected to return a length-m list of strings, where m is the number of events of the second kind. The i^{th} string in this list should correspond to the answer for the i^{th} event of the second kind.

Restrictions

- The following symbols can be used:
  - `list`, `set`, `dict`, `enumerate`, `append`, `pop`, `extend`, `remove`, `sort`, `sorted`, `insert`, `clear`, `reverse`, `reversed`, `iter`, `next`, `zip`.
- The following imports are allowed:
  - `count` and `islice` from `itertools`.
- Loops are allowed.
- Generators and comprehensions are allowed.
- Recursion is *disallowed*.
- Your source code must have at most $2{,}000$ bytes.

Examples

Example 1 Function Call

```
maintain_leaderboard([1, 2, 34], [
  ("submit", ("_daryll_", 2, 2)),
  ("rank", 1),
  ("submit", ("_daryll_", 2, 2)),
  ("submit", ("_daryll_", 2, 2)),
  ("submit", ("_darull_", 1, 34)),
  ("submit", ("_darull_", 3, 34)),
  ("rank", 1),
  ("rank", 2),
  ("submit", ("_daryll_", 3, 42)),
  ("submit", ("_darull_", 1, 1)),
  ("rank", 1),
  ("rank", 2),
])
```

Copy

Example 1 Return Value

```
[
    "_daryll_",
    "_daryll_",
    "_darull_",
    "_darull_",
    "_daryll_",
]
```

Copy

Constraints

- The function `maintain_leaderboard` will be called at most $1{,}000$ times.
- $1 \leq q \leq 5{,}000$
- $0 \leq e \leq 5{,}000$
- The sum of q across all calls to `maintain_leaderboard` will be $\leq 5{,}000$.
- The sum of e across all calls to `maintain_leaderboard` will be $\leq 5{,}000$.
- For each answer a, $|a| \leq 10^{20}$.
- For each query of the first kind:
  - u is a string of ASCII-printable characters with length at most 8.
  - $1 \leq p \leq q$
- For each query of the second kind:
  - $1 \leq k \leq k'$, where k' is the number of unique usernames that have appeared so far.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 75 ❤ points if you solve all test cases where:
  - $q \leq 50$
  - $e \leq 50$
  - The sum of q across all calls to `maintain_leaderboard` will be ≤ 400.
  - The sum of e across all calls to `maintain_leaderboard` will be ≤ 400.
- You get 75 ● points if you solve all test cases where:
  - $q \leq 400$
  - $e \leq 400$
  - The sum of q across all calls to `maintain_leaderboard` will be ≤ 400.
  - The sum of e across all calls to `maintain_leaderboard` will be ≤ 400.
- You get 50 ● points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.