OpenPOTD User's Manual

IcosahedralDice

July 29, 2024

Contents

1 For Contestants

1.1 How to submit

When each new POTD is released, by default the bot will say `DM your answers to me @qotd`. Simply follow the instruction. Answers are the equivalent of a `int64_t` in C++.

1.2 Using the commands

Unless otherwise stated, dates must be provided in ISO8601 format, i.e. `yyyy-mm-dd`. The ID must be a valid problem id, and note that the IDs are not sequential. If the problem is not yet made public (e.g. scheduled to be used as a POTD in the future), then it will be ignored.

The following two commands relate to the scoring interface:

Note 1.1: Commands — Scoring Interface

- `rank` — Displays, by default, a list of the top 20 contestants in the current seasons. To navigate between pages, click the reactions at the bottom of the page. `rank <season_id>` displays the scoreboard for a particular season.
- `score` — Displays, by default, your score for this season. `score <season_id>` displays your score for a particular season. If you are not ranked in the requested season, OpenPOTD will convey this.

The following two commands relate to old problems:

Note 1.2: Commands — Interacting with old POTDs

- `fetch <date_or_id>` — Fetches an old POTD from the database.
- `check <date_or_id>` — Checks the answer for an old POTD. This does not work for the current POTD.

The following three commands relate to privacy. Note none of these commands, by default, change any behaviour of OpenPOTD in Discord. Currently, they only serve1 as a way to communicate with webservers attached to OpenPOTD.

Note 1.3: Commands — Privacy

- `toggle_anon` — Toggles your `anonymous` field in the users field of the database. If a webserver is attached to OpenPOTD, they may choose to use this to not display you on their website.
- `nick <new_nickname>` — Changes your nickname. If a webserver is attached to OpenPOTD, they may choose to use this as your display name on the website.
- `self` — Checks your privacy settings.

1.3 Understanding Scoring

By default, this is the scoring algorithm used by OpenPOTD.
The source code is on the next page.
Let's break it apart. The first section simply fetches all solves from the database and doesn't matter to contestants.

The second section calculates *weighted attempts*. By default, the `weighted_score` function takes 0.9 to the power of the number of attempts `-` 1, i.e. it's 1 if you get it on your first attempt, 0.9 on your second, etc. OpenPOTD then sums all the weighted scores for a each question.

The third section calculates the number of `base_points` each question is worth. By default, this is calculated by taking 1000 divided by the number of weighted solves for each question plus three. In this way, a question with `weighted_solves=47` has `base_points=20`.

The fourth section simply gets all the currently ranked users. Contestants need not worry about this section.

The fifth section calculates the total score of each contestant. It multiplies the base points by your weighted score for each question, and sums them all up.

In this sense, OpenPOTD's grading system rewards contestants for solving hard problems (problems which few others solve), since the total score on a single question has to add up to 1000 in the end. It also rewards contestants who get questions correct on their first try, since each additional try decreases your score by 10%.

Note 1.4: Scoring

```
def update_rankings(self, season: int, potd_id: int = -1):
   cursor = self.bot.db.cursor()

   # Get all solves this season
   cursor.execute('select solves.user, solves.problem_id, solves.num_attempts  from problems left
        join solves '
        'where problems.season = ? and problems.id = solves.problem_id and official = ?', (season,
             True))
   solves = cursor.fetchall()

   # Get weighted attempts for each problem
   weighted_attempts = {}
      for solve in solves:
         if solve[1] in weighted_attempts:
            weighted_attempts[solve[1]] += weighted_score(solve[2])
         else:
            weighted_attempts[solve[1]] = weighted_score(solve[2])

   # Calculate how many points each problem should be worth on the 1st attempt
   problem_points = {i: self.bot.config['base_points'] / weighted_attempts[i] for i in
        weighted_attempts}

   # Get all ranked people
   cursor.execute('select user_id from rankings where season_id = ?', (season,))
   ranked_users = cursor.fetchall()

   # Calculate scores of each person
   total_score = {user[0]: 0 for user in ranked_users}
   for solve in solves:
      total_score[solve[0]] += problem_points[solve[1]] * weighted_score(solve[2])
```

2 For people running OpenPOTD

2.1 Setting Up

The instructions for setting up OpenPOTD are listed in the README file of the git repository.

Detailed instructions for using the Discord interface will follow.