

Submission for Pafu

CF-Suggester

Codeforces practice problem suggester.

A simple web app made with ReactJS and using the Codeforces API. It goes through your codeforces submission history and analyzes it to find out your weak and strong areas. This is done by calculating “strength” values for each problem tag on codeforces. Strength values can be considered roughly on par with rating.

So if a user has a strength value of 1600 for the dp tag, it means that said user should be able to solve problems tagged with dp of rating around 1600. Initial strength values are set to 1000 for all tags.

The Codeforces API has an endpoint for accessing a users submissions in json format. After making a request to this endpoint, submission data is received. Submissions are iterated in order from earliest to latest. For each correct submission, the strength value for the tags associated with the problem to which the submission was made changes. The change depends on whether the submission was correct or not, the difficulty of the problem, current strength of the user for a tag. To calculate changes, two special functions are used - one for correct submissions and one for incorrect submissions. (Time limit errors, runtime errors, compilation errors, etc. are all considered incorrect submissions).

For incorrect submissions, the following function is used:

$$s_{t+1} = \min\left(\max\left(s_t - 100, \sqrt{s_t d}\right), s_t + 50\right) - 50$$

Here:

s_t is the strength rating after iterating over t submissions

d is rating of problem.

Basically, the function gives the new strength value s_{t+1} if a user with strength s_t solves a question with difficulty d incorrectly.

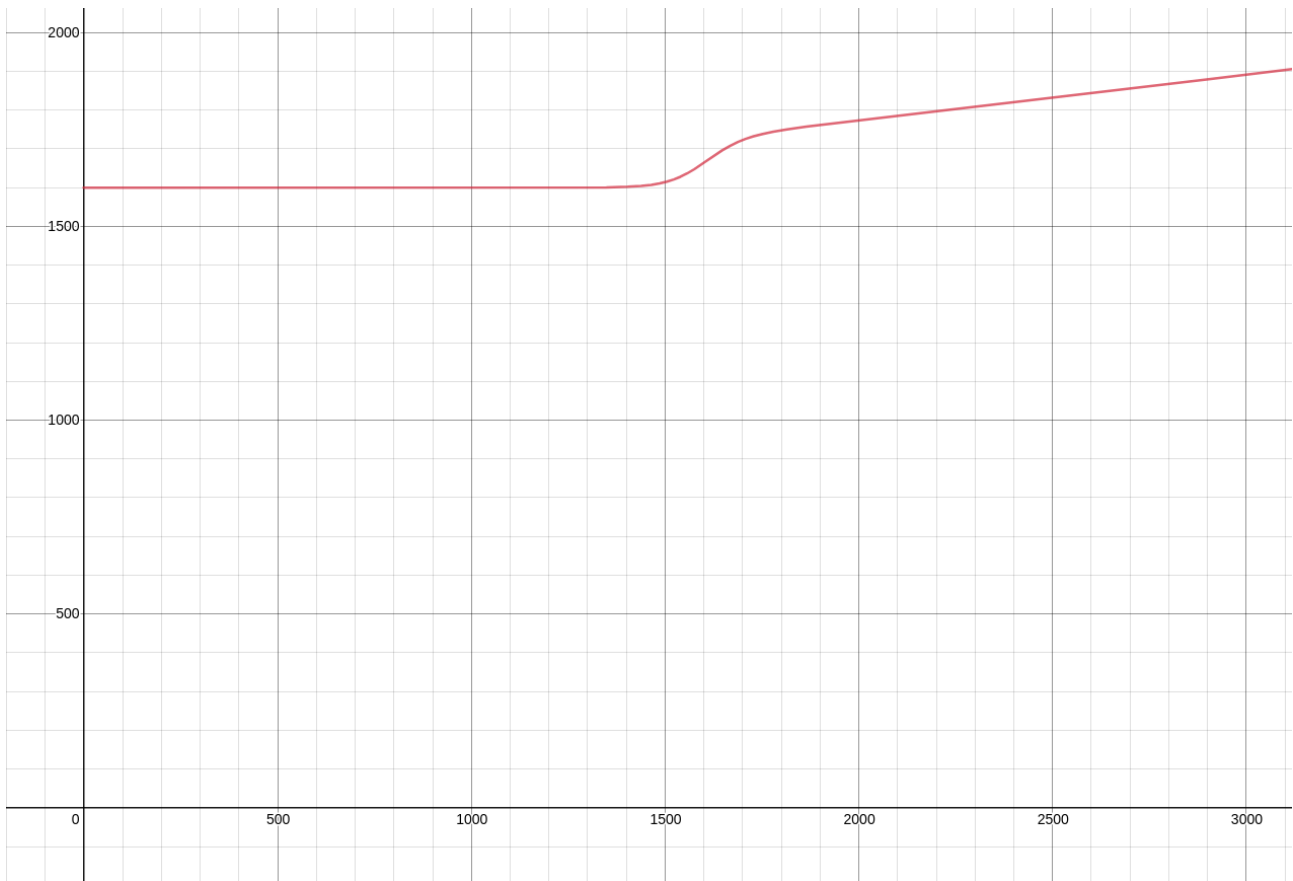
This function ensures that maximum strength drop is at most 150 points. It also makes sure that strength rating drops only when an incorrect submission is made on problems with rating less than or equal to current rating.

For correct submissions, the following function is used:

$$s_{t+1} = s_t + \left(\frac{50 \cdot d}{s_t} - 10\right) \cdot \frac{\exp\left(\frac{d - s_t}{50}\right)}{\exp\left(\frac{d - s_t}{50}\right) + 1} \cdot \log_{10}(d + 1)$$

The choice of this function was mostly ad-hoc, based on looking at graphs and intuition but we're fairly confident that this function has many desirable properties.

Taking a look at the plot for s_{t+1} versus d for $s_t = 1600$ can help understand why this function is a good choice.



As can be seen, for successfully solving problems with d much less than s_t , strength increase is negligible. As d nears s_t , strength increase becomes more significant. For d greater than s_t , till a certain value, strength increase is at a steady rate and after that, the slope drops off. The drop off is to ensure that solving a single problem much higher than your strength rating doesn't artificially inflate your strength rating. Only after multiple successful submissions at a particular rating d does the strength value reach close to d .

After going over all submissions, a list of tags and their corresponding strengths is known. This list is further narrowed down by only considering tags that have 5 submissions at least. From these tags, the 5 with lowest strength are picked.

The Codeforces API also has an endpoint to return a list of problems by tag. For each of the weak tags, a list of problems is fetched from codeforces. This list is then filtered to only include problems with rating between 80% to 120% of the strength for that tag. This is done so that neither too easy nor too difficult problems are recommended. From that list, 5 problems are picked at random (for some users and some tags, less than 5 problems may be available depending on calculated strength and actual number of problems on codeforces for that tag). The problems, thus selected for all weak tags are then displayed to the user as links.

Some screenshots:

