Optional Contest: Hog Strategy

hog_contest.zip (hog_contest.zip)

A strategy is
A single function that makes
Ten thousand choices.

Instructions

> This contest is completely optional!

Download a blank `hog_contest.py` file and simple tests for correct formatting as a zip archive (hog_contest.zip). Type `python3 ok` to run the provided tests.

Submit a `hog_contest.py` file containing a function called `final_strategy` and a `PLAYER_NAME`.

```
python3 ok --submit
```

The contest ends on **Friday, February 22th at 11:59 PM**. We will use your latest submission before this date to determine the final results of the contest.

Game rules

To make things more interesting, we've introduced a new rule (**Time Trot**) in addition to the original set of special rules. Here is the entire set of rules:

- **Time Trot**. A turn involves a player rolling dice, and each turn is numbered, starting from 0. If a player chooses to roll a number of dice `k` on turn `n`, and `n % 5 == k`, then that player gets an extra turn immediately after the current turn. However, a player cannot get an extra turn immediately after an extra turn.
  - *Note*: A strategy must be a deterministic function of the players' scores and cannot track the turn number or previous actions. See complete contest rules below.

Contest rules

Teams can have one or two people. Each person can only be part of one team.

Each submitted strategy will play against all other submissions. The player to go first will be determined by a flip of a fair coin. We will exactly compute the expected win rate for each player, so that the outcome of this tournament will be determined by strategy alone and not the roll of the dice or flip of a coin. A submission scores a match point each time it has an expected win rate above 50.0001%. We will rank submissions based on the number of matches they won. Ties count as losses.

The top three submissions will earn the following:

1. First place gets 3 points of extra credit.
2. Second place gets 2 points of extra credit.
3. Third place gets 1 point of extra credit.

Winners will also be publicly recognized in future iterations of 61A!

Some additional contest rules:

- Calling your `final_strategy` function on every possible pair of scores should take less than 10 seconds. The provided tests check for this.
- All strategies must be deterministic, pure functions of the current player scores! Non-deterministic strategies or strategies based on the history of the game will be disqualified.
- Please don't do anything nefarious to learn about or influence other students' submissions.

If you have any questions about the rules, don't hesitate to post on Piazza.

Happy coding!

CS 61A (/)

Weekly Schedule (/weekly.html)

Office Hours (/office-hours.html)

Staff (/staff.html)

Resources (/resources.html)

Studying Guide (/articles/studying.html)

Debugging Guide (/articles/debugging.html)

Composition Guide (/articles/composition.html)

Policies (/articles/about.html)

Assignments (/articles/about.html#assignments)

Exams (/articles/about.html#exams)

Grading (/articles/about.html#grading)