# 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 `final_strategy.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 `final_strategy.py` file containing a function called `final_strategy` and a `PLAYER_NAME`.

```
python3 ok --submit
```

The contest ends on **Thursday, July 8 at 11:59 PM**. We will use your latest submission before this date to determine the final results of the contest.

You can use the `compare_strategies.py` file to compute the exact win rate between any pair of two strategies using our server. Depending on server load, we may rate-limit or disable this functionality at any point, so consider re-implementing this file yourself locally!

Your strategy can run as long as you want. If `ok` complains that you're exceeding the time limit, run it with the flag `--timeout 0` to disable the time limit.

## Leaderboard

The leaderboard can be found here (https://hog-contest.cs61a.org). It will continuously update as the contest progresses. Go to the server log (https://hog-contest.cs61a.org/log) to see the status of your submission.

## 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 strictly 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.

## 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 are the full set of rules:

- **Sow Sad**. If any of the dice outcomes is a 1, the current player's score for the turn is 1.
- **Piggy Points**. A player who chooses to roll zero dice scores `k + 4` points, where `k` is the absolute value of the tens digit minus the ones digit in the opponent's score. If the opponent's score is only one digit, assume the tens digit is 0. You may **not** assume that the score is under `100`.
- **More Boar**. After the points for the turn are added to the current player’s score, the current player takes an extra turn if the minimum digit of the current player's score is strictly *smaller* than the minimum digit of the opponent's score **and** the maximum digit of the current player's score is strictly *larger* than the maximum digit of the opponent's score. You may **not** assume that the scores are under `100`. The More Boar calculation should be done on the current player's score **after** the points from the current turn are added. More Boar can be activated multiple times in a row for the same player (see Example 4).
- **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 % 8 == k`, then that player gets an extra turn immediately after the current turn. However, a player cannot get an extra turn as a consequence of Time Trot immediately after an extra turn, though they can get an extra turn as a consequence of More Boar. Furthermore, players will roll 8-sided dice on all extra turns (including those produced from More Boar).

## Submission rules

- 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 not behave as expected when submitted and will likely not do well.
- You may use any libraries (such as NumPy, TensorFlow, or anything else you can think of) in your submission.

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

Happy coding!