

16 марта 2024

#DEVELOBEAR

VII Турнир по программированию ЯрГУ и НПО Криста



#DEVELOBEAR2024

Правила игры

Вы – группа учёных – работаете над исследованием очень перспективной технологии. Однако у вас есть конкуренты – другие научные лаборатории, которые тоже исследуют эту технологию. Вы понимаете, что в современном научном мире никто не работает в одиночку. Поэтому пытаетесь взаимодействовать с другими лабораториями. В процессе сотрудничества можно обмениваться как настоящей информацией, помогающей обеим лабораториям продвигаться в исследовании, либо дезинформацией, которая не даёт прогресса для конкурирующей лаборатории и даже откидывает её назад.

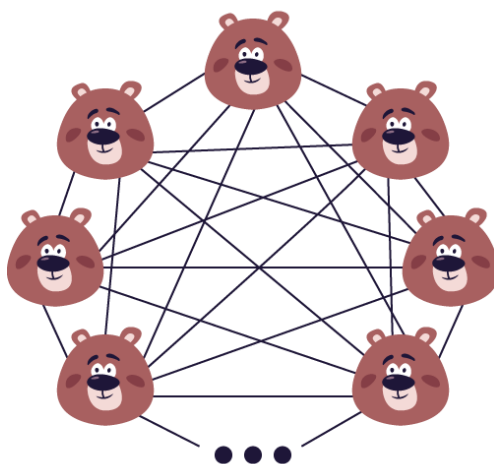
Балансируя между этими крайностями, нужно стать лидером исследования.

Состязания будут проходить в несколько туров. В первом туре каждая группа учёных играет с каждой группой учёных-соперников. На второй тур переходят 60 лучших групп, и опять играют каждый с каждым. На третий тур переходят 40 групп. В финальном туре будет 20 групп, между которыми и будут определены победители, которые продвинулись в исследовании технологии больше остальных.

Тур 1	Все команды
Тур 2	60 лучших
Тур 3	40 лучших
Финальный тур	20 лучших

Все очки исследования, заработанные в каждом туре, переходят в следующий.

В рамках тура необходимо провести серию экспериментов с каждой группой.



В рамках одной серии экспериментов проводятся 30 экспериментов совместно с одной из групп учёных.

После каждого эксперимента вы должны решить, какие результаты передавать конкурирующей группе учёных – **правдивые** или **ложные**.

При этом конкурирующая группа учёных также будет решать, какие результаты передать вам – правдивые или ложные. После каждой передачи результатов вы узнаете правдивые были сведения или ложные.

Тур	Все серии экспериментов каждого игрока с каждым
Серия	Серия экспериментов с одним соперником. В рамках одной серии проводятся 30 экспериментов
Эксперимент	Один эксперимент в рамках серии с одним соперником. Результатом является решение передать правдивые сведения или ложные

Расчёт очков

В рамках одной серии экспериментов вы проводите 30 экспериментов с одной командой. Точнее, вы должны принять решение какую информацию об эксперименте отдать конкурирующей группе – **правдивую** или **ложную**.

Если вы решили передать **правдивые** результаты, и соперник решил передать **правдивый** результат, то вы оба получаете по 2 очка.

Если вы решили передать **ложный** результаты, и соперник решил передать ложный результат, то вы оба получаете по 0 очков.

Если вы передали **ложный** результат, а соперник – **правдивый**, то вы получаете преимущество в 3 очка, а соперник откатывается назад -1 очко.

И наоборот, если вы передали правдивый, а соперник ложный результат, т.е. обманул вас, то вы получите -1 очко, а соперник получит 3 очка.

Матрица очков

Группа 1 \ Группа 2	Правда	Ложь
Правда	2 \ 2	-1 \ 3
Ложь	3 \ -1	0 \ 0

Определение победителя

Побеждает группа, набравшая максимальное количество очков по результатам всех туров.

Входные данные

- `answersEnemy` – логический массив с ответами соперника в рамках текущей серии экспериментов
- `answersOwn` – логический массив с вашими ответами в рамках текущей серии экспериментов
- `ownScore` – ваше текущее количество очков за всё время (не включая текущую серию экспериментов)
- `enemyScore` – текущее количество очков противника за всё время (не включая текущую серию экспериментов)
- `allSeries` – количество серий, проведённых вами с соперниками, во всех турах
- `allScores` – массив с количеством очков всех игроков на текущий момент
- `currentExperiment` – номер эксперимента в текущей серии
- `totalExperiments` – всего экспериментов, которые надо будет провести в рамках текущей серии, можно считать константой 30

Выходные данные

Результат работы вашего алгоритма – одно логическое значение `true` или `false`, которое означает передачу сопернику правдивых сведений (`true`) или ложных (`false`).

Задача алгоритма

Задача вашего алгоритма – принять решение, возвращать сопернику настоящие данные или поддельные. Для этого можно анализировать свои шаги и шаги соперника в рамках текущей сессии экспериментов. Также можно использовать знание о продвижении соперников в исследовании относительно вас.

Особенности

Существует ограничение времени выполнения алгоритма, однако его даётся более чем достаточно. Ограничение носит защитный характер от закливания. Если алгоритм выходит за таймаут, то ему начисляется -1 очко за текущий эксперимент.

Среда разработки

Вам дана функция, куда передаются все эти параметры и которая должна вернуть логическое значение – true или false. В данном случае возвращается true или false с вероятностью 50%.

```
function alg1({
  ownScore,
  enemyScore,
  allSeries,
  allScores,
  answersOwn,
  answersEnemy,
  currentExperiment,
  totalExperiments,
}: StepArgs) {
  return Math.random() < 0.5;
}
```

Для моделирования игрового процесса вы можете написать несколько таких функций и реализовать разные алгоритмы поведения, просто, назвав их по-разному, например, alg2, alg3 и т.п.

Для запуска этих алгоритмов на игру каждого с каждым их достаточно передать в строке в конце страницы с кодом:

```
export default [alg1, alg1, alg1, alg2, alg3]
```

Такая строка означает, что будет запущена на исполнение 5 игроками: первые трое будут играть по алгоритму, заложенному в функции alg1, четвертый игрок – по алгоритму из alg2, пятый игрок – по алгоритму из функции alg3.

Таким образом, вы можете заложить предполагаемое поведение других игроков и своё, попробовать посмотреть, кто из вас победит.

Эту строку вы можете отправить и на турнир. При запуске на турнире будет взят первый указанный в массиве алгоритм, и именно он будет выступать от вас. Поэтому, чтобы не забыть, лучше свой основной алгоритм всегда размещать первым в этом массиве.