



Swiss Stig (Part 1)

Some say, unlike his silent cousins which have to rely on body expressions, he could react to questions with  and  on Discord.



Some say, he was once caught after a Codelt Suisse event for drunk driving, as the event organizers used to distribute alcohol drinks when the event was conducted offline.

He's not the Stig, but he's the Stig's Swiss cousin.

Important message

This is a multi-part challenge. You may directly reuse solutions of later parts to solve earlier parts for full score.



Background

The car show *Toxic Gas* has just commissioned the Swiss Stig to benchmark a new model, and is asking them to rate the car using a scale from 1 (terrible) to MAX (perfect). The Swiss Stig, being a Stig, could not speak but only react in  and . Given the complexity of figuring out Stig's opinion on the model, the producer has assigned a (self-proclaimed) PhD candidate to interview the Swiss Stig -- the master interviewer Jeremy.

Before you shout "Binary search!", Jeremy has invented several suboptimal interview methods to fuel your nightmares over this hackathon night.

1. Simply repeat the following process Q times

Jeremy: Is the rating between $\$from$ and $\$to$?

Stig:  (or  whenever appropriate)

After all Q questions are asked, we could devise a non-empty set of ratings that is consistent with the Stig's replies. Jeremy will then pick one of the values uniformly at random and assume that's the Stig's actual rating.

Another host of the show, the master philosopher James, is asking the following question

Assuming underlying distribution of the car's rating is also uniformly at r

what is the probability that Jeremy nails his guess correctly, i.e., his ac

Inputs

The input json contains a list of "Interview" items, which contains a "questions" attribute describing what Jeremy asked.

```
[{
  "questions": [[
    {
      "from": Int,
      "to": Int,
    }: Interval
  ]: List<Interval> as Question
]: List<Question>,
"coefficients": [{
  "p": 0,
  "q": 0,
}: Coefficient
]: List<Coefficient>,
"maxRating": Int
}: Interview
]: List<Interview>
```

It is guaranteed that $1 \leq \text{from} \leq \text{to} \leq \text{maxRating}$, and each Question contains exactly one interval.

For now, ignore the "coefficients" attribute which is reserved for other subtasks.

Output

```
[{
  p: Int,
  q: Int,
}: Result
]: List<Result>
```

For each of the interviews, output a "Result" object that answers Jame's question, which satisfies the conditions of ...

1. p/q is an irreducible fraction
2. $p \leq q \leq \text{MAX}$
3. No other fraction that satisfies the first two conditions gives a better estimation on James' question (in regard to this Interview instance)

Constraints

For ALL subtasks...

1. There will be 5 interviews per subtask.
2. $Q \leq 10^5$
3. $\text{maxRating} = 10^9$
4. Time limit: 15 seconds

Scoring

Subtask total: 400

For ALL subtasks, each interview contributes 20% of the set's sub-total. The score is awarded if and only if both p and q are exact.

Endpoints

Expose an endpoint `/stig/perry`