MENU

FIND MEMBERS BY USERNAME OR SKILL

survival07

EDIT

DASHBOARD

MY PROFILE

PAYMENTS

SETTINGS

LOG OUT

COMPETE

DESIGN CHALLENGES

DEVELOPMENT CHALLENGES

DATA SCIENCE CHALLENGES

COMPETITIVE PROGRAMMING

LEARN

GET STARTED

DESIGN

DEVELOPMENT

DATA SCIENCE

COMPETITIVE PROGRAMMING

COMMUNITY

OVERVIEW

TCO

PROGRAMS

FORUMS

Support

Forums

Re: Post your approach (response to post by **sullyper**) | Feedback: (+1/-0) | [+] [-] | Reply

2 edits | Sat, Nov 7, 2020 at 11:49 AM BDT

dimkadimon
4478 posts

Sounds like problems with your ILP. I think it gave you issues in the past too?

In the end 1% may not make any difference to your final ranking as you are 1.5% behind the next place.

Re: Post your approach (response to post by **dimkadimon**) | Feedback: (+1/-0) | [+] [-] | Reply

1 edit | Sat, Nov 7, 2020 at 12:10 PM BDT

sullyper
264 posts

Thanks, but yes I can see all the seeds and the log of my program. So I know which seeds I failed.
Anyway, I would say lesson learned but apparently I never learn =)

And to answer your question, it's the first time I use my ILP in a competition (and maybe last...), So no never had an issue before.

Where do you see I am 1.5% behind the next place? I see a very small difference between **yowa** and myself, anyway, it does not matter, the failures are real, so I get the ranking I deserve for my sloppiness!

Re: Post your approach (response to post by **ika**) | Feedback: (+1/-0) | [+] [-] | Reply

Sat, Nov 7, 2020 at 10:23 PM BDT

nika
370 posts

Similar solution with two phases
1. Predicting original attack and defense strengths using SA that minimizes the product of probabilities of each player ending up with given number of scored and conceded goals. For given strengths we have the probabilities of each score happening (very similar to **eulerscheZahl**'s code except the last part where I have separate probability for each goal count instead)
2. We haven't used the points information yet. We can refine initial probabilities using the Bayes theorem with dynamic programming like described in **ika**'s post. To fit n=50 in time/memory I had to cut out the most important 16x16x50 sub-array on each step. The last improvements were to repeat this step once

again (which gave around +0.3% locally), taking the weighted average of probabilities (0.2*old+0.8*new) and in the end return the score that maximizes 2*score+1*outcome (another +0.3%), not the most likely one.

/