COMPETITIVE PROGRAMMING PROBLEM RECOMMENDATION SYSTEM

Final Presentation: AI 705

umm actually

BACKGROUND

Competitive Programming:

A mind sport of solving algorithmic problems under time constraints, popular among software engineers and coding enthusiasts.

Codeforces:

A renowned online platform for competitive programming.

CODEFORCES RANKS

- **Contests:** Programming contests where participants solve algorithmic problems within specified time limits.
- **Ratings:** Assigned to participants based on their performance in contests, reflecting their skill level and progress in competitive programming

Rating Bounds	Color	Title
≥ 3000	Red	Legendary Grandmaster
2600 — 2999	Red	International Grandmaster
2400 — 2599	Red	Grandmaster
2300 — 2399	Orange	International Master
2100 — 2299	Orange	Master
1900 — 2099	Violet	Candidate Master
1600 — 1899	Blue	Expert
1400 — 1599	Cyan	Specialist
1200 — 1399	Green	Pupil
≤ 1199	Gray	Newbie

THE PROBLEM

Programmers waste hours searching for good problems to solve.
Currently there is no personalised recommendation system for competitive programming!

THE SOLUTION

Develop a recommendation system with the goal of increasing a users rating!

Recommend problems tailored to a user’s skill level and progress.

Empower users to focus on the most impactful practice problems, accelerating their development as competitive programmers

CHALLENGES

- No user feedback for problems.
- How to decide whether a problem is “good” for a user?
- Just because a user likes a problem, doesn’t mean that the problem is good for the user and vice versa.
- **Solution**: Super Users

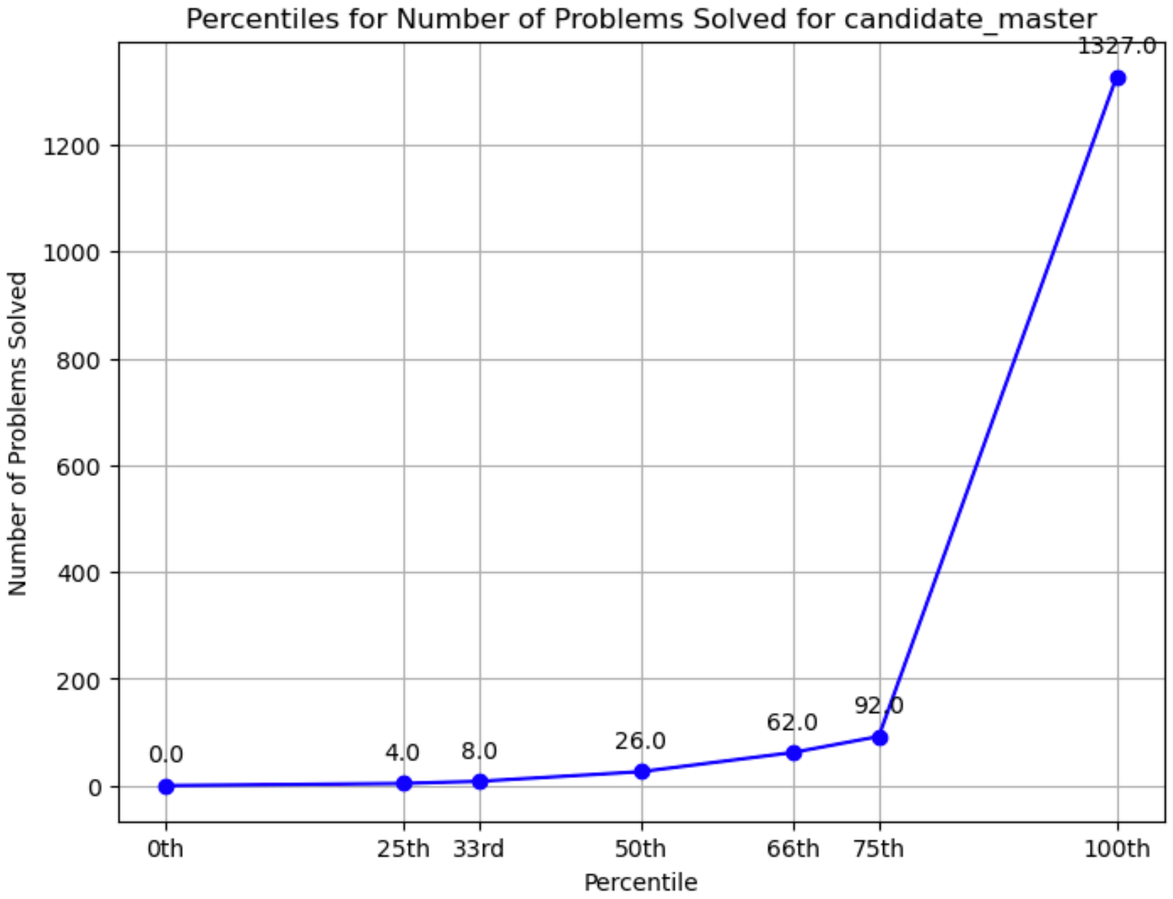
SUPER USERS

- Successful users of the platform.
  - They improved their rank in a short span of time
  - They have solved a decent number of practice problems in that span of time.

- Motivation hypothesis: Super Users are practicing the right problems.
  The widespread popularity of A2OJ and C2 ladders supports this claim

- Idea: Recommend problems that are solved by super users.

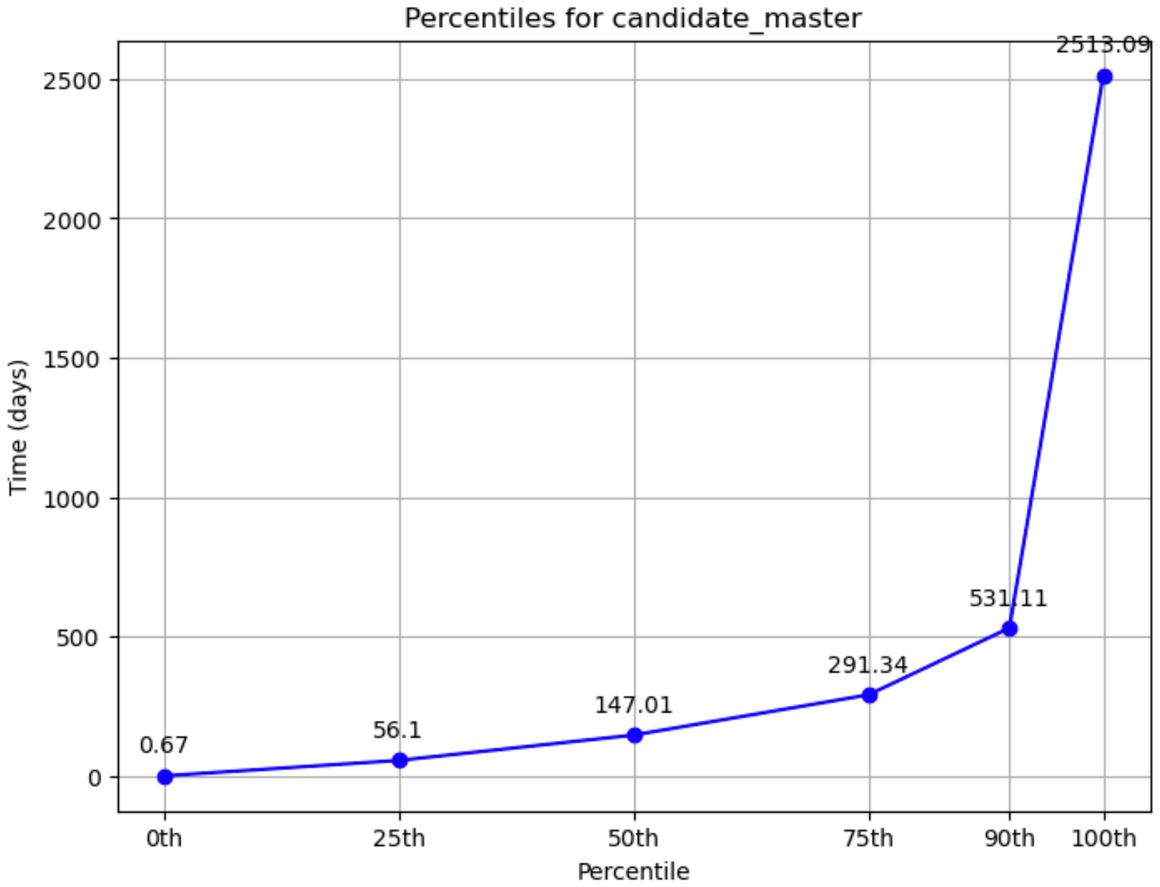
IDENTIFICATION OF SUPER USERS



Filtering out alt accounts:

- Users who have obtained ratings without practising are not taken into account.
- Typically, these are secondary or even tertiary accounts of an individual.
- Therefore, we exclude users who have solved minimal problems between changes in rank.
- For example, we have chosen the 66th percentile for this ranking threshold.

IDENTIFICATION OF SUPER USERS



Finding “super” users

- We aimed to discover users who rapidly accumulated ratings.
- We select low percentiles for this group metric, but high enough to ensure that a reasonable amount of users are included.
- For example, we chose the 50th percentile as the benchmark for this ranking.

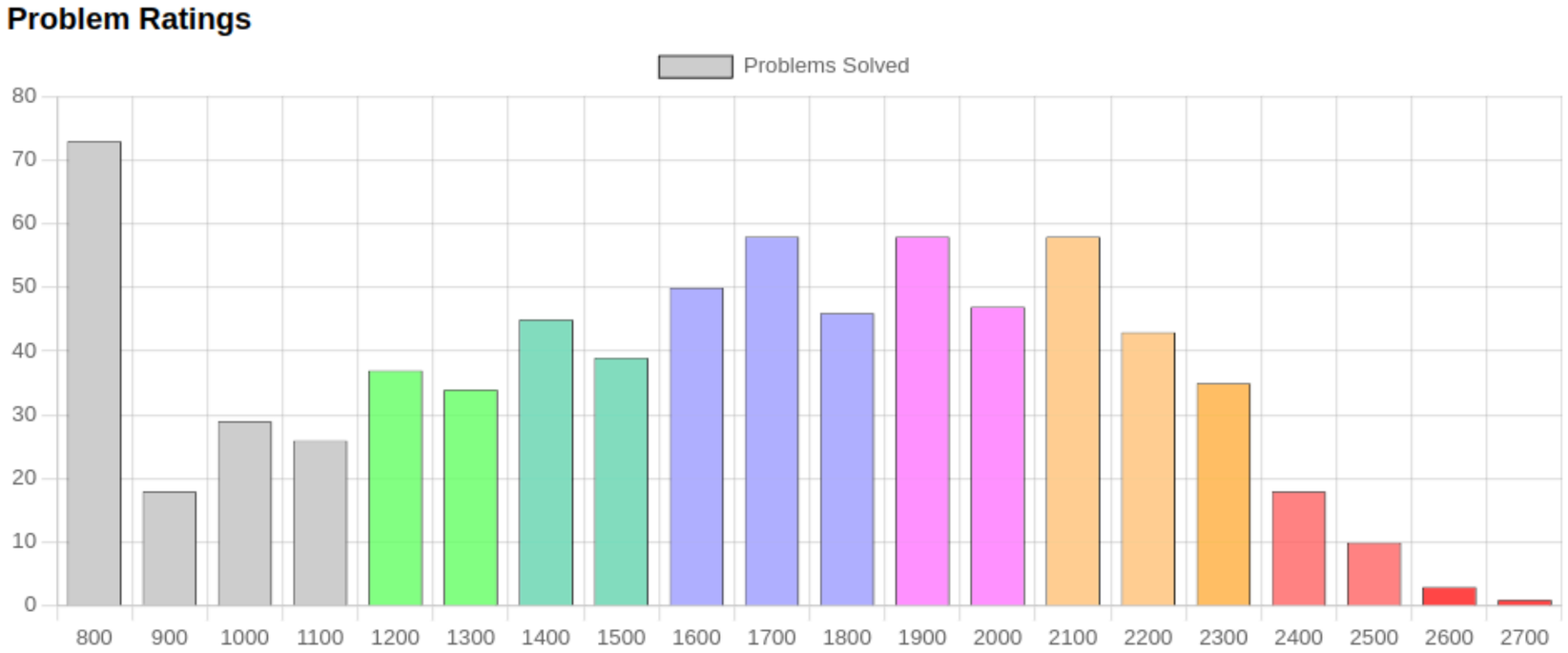
DATASET CURATION

- Following our EDA, we developed a script utilizing the Codeforces API to identify and extract super users within a specified rank based on predefined constraints.
- For each identified super user, we retrieved a comprehensive list of problems they practiced during the specified time period, including timestamps indicating when each problem was solved.
- The above data was in JSON format, so we wrote another script to convert it to CSV

user_handle	problem_id	timestamp
rgnerdplayer	1538:D	1626961617
devinqu	1538:C	1626959972
akua	1538:B	1626959819
ahsoltan	1538:A	1626959622
Suwan	1520:G	1626959357
Fyind	1520:F1	1626958453
fzx	1520:F2	1626958333
sg78276397	1520:E	1626956581
Fork512Hz	1520:D	1626956431
efishel	1520:C	1626956335
BowTen	1520:B	1626955926
Tx_Lcy	1520:A	1626955812

DATASET CURATION

- Additionally, we collected supplementary user data detailing the distribution of difficulty levels of problems that each user has solved.

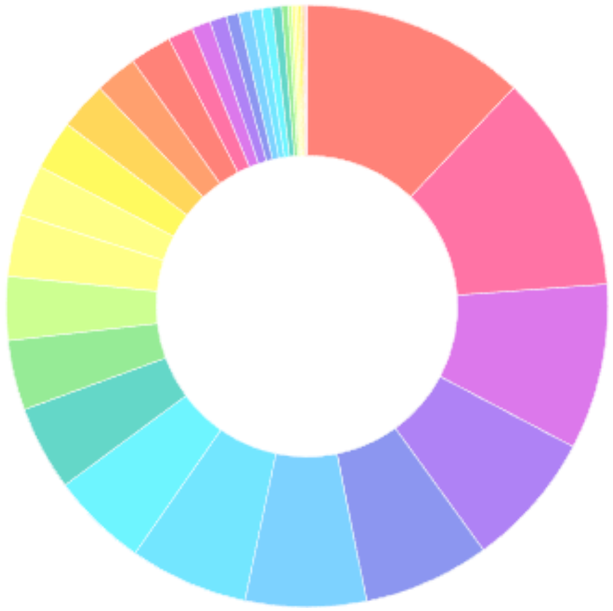


user_handle	1000	1100	1200	1300
maspy	13	13	23	23
wsyear	15	6	17	8
LXH-cat	16	13	17	12
skittles1412	3	1	3	1
PurpleCrayon	31	22	27	31
maomao90	5	7	17	12
xzc0920	8	4	4	2
AmirAli-Asgari	2	13	5	3
Dominater069	14	10	21	16

DATASET CURATION

- We also collected user data detailing the distribution of topics they've tackled, including categories like ad-hoc, mathematics, binary search, and more.

Tags Solved



- greedy : 286
- math : 279
- implementation : 211
- dp : 173
- brute force : 161
- constructive algorithms : 154
- data structures : 151
- sortings : 123
- binary search : 108
- number theory : 89
- dfs and similar : 81
- graphs : 79
- trees : 65
- combinatorics : 63
- two pointers : 61
- bitmasks : 54
- strings : 53
- dsu : 32

user_handle	binary search	bitmasks	brute force
maspy	32	11	56
wsyear	25	15	47
LXH-cat	51	23	84
skittles1412	3	1	5
PurpleCrayon	30	28	77
maomao90	11	4	24
xzc0920	18	8	20
AmirAli-Asgari	1	4	18
Dominater069	19	8	41

THE RECOMMENDER MODEL

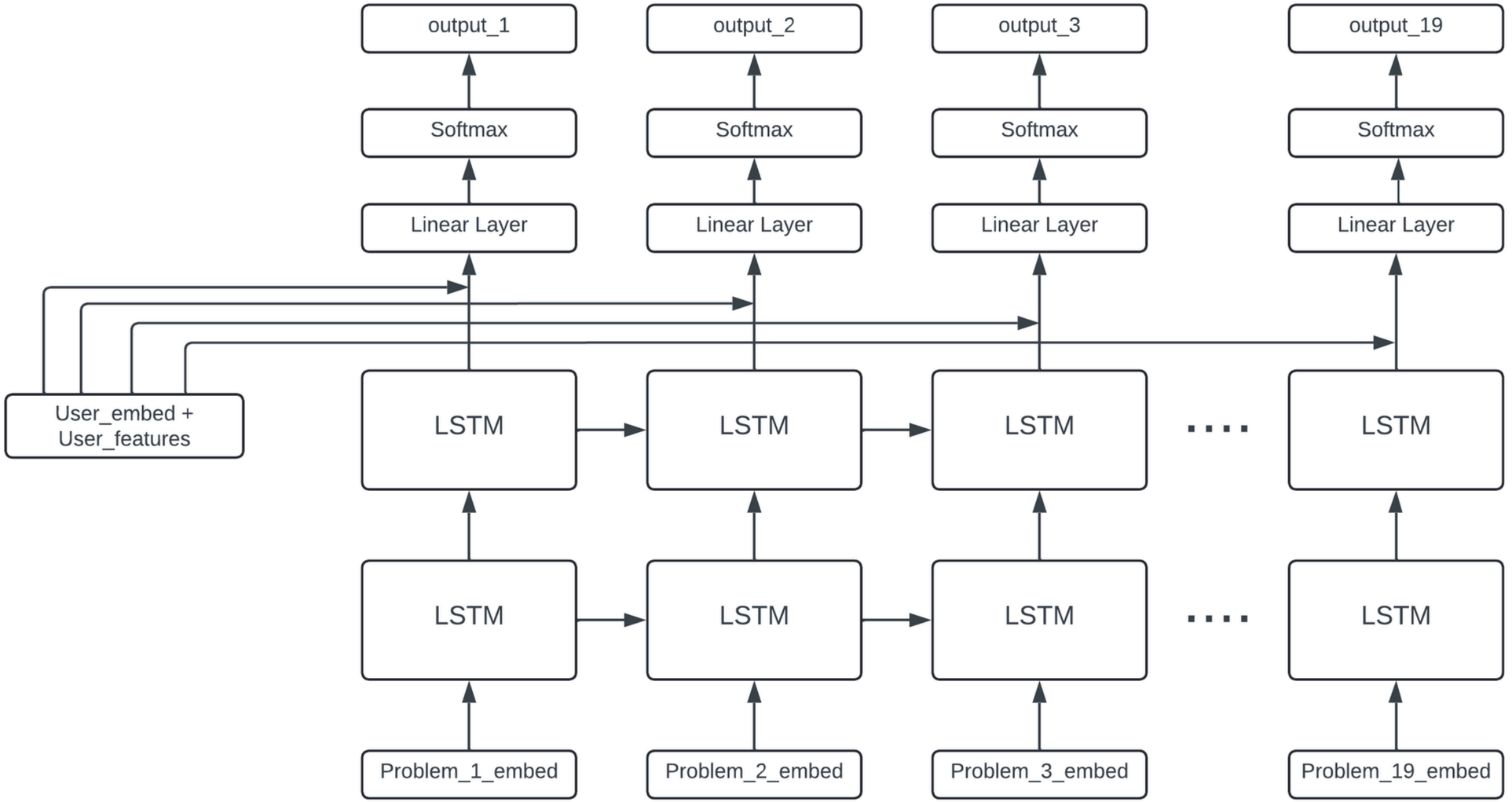
- **Core ideas:**
  - Problems solved by super users directly contributed to their success.
  - Sequence (order) in which problems are solved matters.

- Based on the above two points, we decided to implement a Seq2Seq approach of predicting the next problem a user to solve.

- Our model of choice: Long Short-Term Memory (LSTM)

- For each super-user, we created timestamp-sorted sequences of the problems they have solved. This data, along with the user features(tags and ratings distribution) is used to train the LSTM.

THE RECOMMENDER MODEL



THE RECOMMENDER MODEL

Hyperparameters

- Number of LSTM layers: 2
- Dropout rate (to prevent over-fitting): 0.2
- Embedding dimension: 128
- GRU hidden layers dimension: 128
- Learning rate: 1.0
- Loss function: Cross Entropy
- Optimiser: Stochastic Gradient Descent
- Number of epochs = 30 (with early stopping)

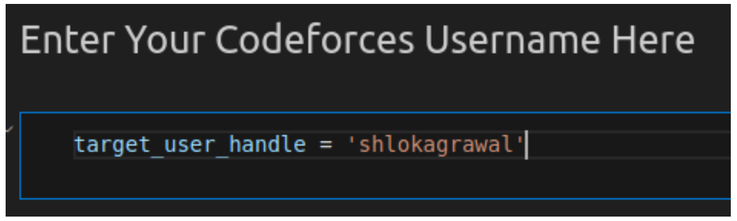
THE RECOMMENDER MODEL

During Inference:

- The user enters their Codeforces username
- Calls are made to the Codeforces API to get the users data
- Data is processed and fed into the model
- Model outputs the top 10 problems that the user should solve next.

EXAMPLE

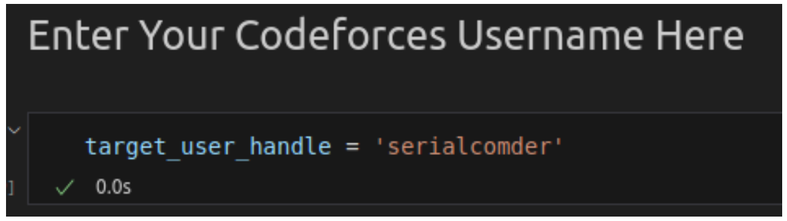
Input



Output

```
Recomendations:
-problem_104687:J
-problem_1812:F
-problem_999:F
-problem_1392:B
-problem_842:C
-problem_1243:B2
-problem_1169:A
-problem_717:C
-problem_1427:C
-problem_1428:E
```

Input



Output

```
Recomendations:
-problem_1111:C
-problem_527:B
-problem_1634:E
-problem_1029:A
-problem_501:B
-problem_1447:D
-problem_1695:B
-problem_1929:F
-problem_1051:F
-problem_1194:E
```

THANK YOU

Questions?