HAC: The Hacker-Cup AI Competition

Weiwei Yang[*] Mark Saroufim Joe Isaacson Luca Antiga Greg Bowyer
Driss Guessous Christian Puhrsch Geeta Chauhan Supriya Rao Margaret Li
David Harmeyer Wesley May

Abstract

We are launching the first AI track for the popular Meta Hacker Cup programming competition, designed to assess the capabilities of Generative AI in performing autonomous code generation tasks. We aim to test the limits of AI in complex coding challenges and measure the performance gap between AI systems and human programmers. We will provide access to all Hacker Cup problems since 2011 alongside their respective solutions in a multimodal (image and text) format, and utilize the existing Hacker Cup infrastructure for competitor evaluation. Featuring both "open evaluation, open model" and "open evaluation, closed model" tracks, this competition invites diverse participation from research institutions of varied interests and resource constraints, including academic labs, AI startups, large technology companies, and AI enthusiasts. Our goal is to develop and democratize meaningful advancements in code automation with the very first open evaluation process for competitive AI programmers.

Keywords

Generative AI, Code Generation, AI Agent, AI system integration, Efficiency

1 Competition description

1.1 Background and impact

Generative AI has shown significant promise in code generation, with tools such as GitHub's Copilot and Amazon CodeWhisperer finding widespread adoption and integration into development environments to enhance programmer productivity. This phenomenon is the result of a recent emergence of language models (LMs) capable of generating code, whether specialized, such as OpenAI's Codex(Chen et al., 2021), Salesforce's CodeT5(Wang et al., 2021), and Meta's Code Llama (Rozière et al., 2024), or general-purpose models with coding generation like Mistral 7B(Jiang et al., 2023) and Phi(Li et al., 2023).

A natural arena for evaluating such code generation models, competitive programming has a long history as a mind sport that tests reasoning, abstraction, mastery of symbolic languages, critical thinking, and the ability to perform under time pressure. Competitions like Hacker Cup, Codeforces, and TopCoder pit contestants against one another to solve challenging problems. The interest in applying AI to coding competitions has grown, as seen with systems like AlphaCode(Li et al., 2022) and AlphaCodium(Ridnik et al., 2024) participating in events like Codeforces. However, no dedicated AI track has been established yet in these competitions, representing a missed opportunity for advancing AI in code generation.

As organizers, we pursue a goal similar to our 2023 NeurIPS LLM Efficiency Challenge: furthering the democratization of AI and encouraging the development of human-level AI programming ability.

[*] Corresponding organizer. weiwei.yang@microsoft.com

Preprint. Under review.

To this end, we are introducing a dedicated AI track in the Meta Hacker Cup competition at NeurIPS. Meta Hacker Cup, formerly known as Facebook Hacker Cup, is an annual international programming competition that has attracted thousands of competitors since 2011, who are rewarded with prestige and cash prizes. The competition spans five rounds over several months, with live streams providing commentary and results.

The AI track will follow a format consistent with the human track, with five rounds of coding problems of increasing difficulty presented across two months from mid-September to early December. Human and AI tracks will receive the same challenge problems at the same time. Human and AI participants will be ranked independently within their tracks by their correctness and include partial credit. Two parallel AI tracks are planned: an "open evaluation, open model" track limited to a single A100 GPU, building on learnings from the 2023 NeurIPS LLM Efficiency Fine-tuning Challenge; and an "open evaluation, closed model" track where challenge questions are submitted to closed model hosts. This dual-track approach balances the goals of making AI accessible to those with limited resources and significantly advancing AI code generation capabilities.

Effective competitive coding AI systems need to understand specifications in natural language, read diagrams, design algorithms, generate code, interact with runtimes, and debug. This comprehensive skill set exceeds the abilities of current foundational models - for example, GPT-4(OpenAI et al., 2024) could only solve the simplest 2023 Hacker Cup practice problem using a basic zero-shot chain-of-thought prompting strategy 4.1. A performant system may require combining specialized coding techniques such as adaptable pretraining, data curation, efficient fine-tuning, retrieval augmentation, tool use, multi-agent approaches, and advanced prompting - all areas of active research in the AI community.

This competition welcomes diverse participants, from well-resourced institutions in the closed model track to enthusiasts in the open model track. Unique features like ranking AIs alongside top competitive coders and live commentary are expected to generate significant interest. Tutorials and guest lectures will help onboard participants, especially younger enthusiasts, aiming to inspire the next generation of AI researchers and developers. Similar to the 2023 NeurIPS LLM efficiency fine-tuning competition, this event will distill reproducible insights to provide a foundation for the community to build their own code generation systems.

1.2 Novelty

While AlphaCode has previously entered Codeforces (Li et al., 2022), the absence of a dedicated AI track in a major international coding contest, with multiple AI Agents simultaneously competing, marks a significant missed opportunity. This gap has limited the exposure and participation of a wide population, restricting the potential for communal growth and innovation in AI-driven coding solutions.

Moreover, the challenge of creating competitive AI systems on constrained hardware has been largely unaddressed. The specifics of AlphaCode's hardware utilization remain undisclosed by its creators, DeepMind, but it's widely presumed to require multiple GPUs. This assumption highlights a notable gap, the absence of a known system capable of competing on a single GPU. Our competition seeks to bridge this divide with the "Open Evaluation, Open Model," explicitly designed to foster innovation within resource constraints, which both pushes to democratize AI and foster research into resource-efficient environmentally conscious AI.

To succeed, entries need to push beyond current code generation models towards autonomous code generation systems. This may require innovation in areas such as efficient model architectures, knowledge composition, such as Mixture of Experts (MoE) (Jiang et al., 2024) and model merging (Akiba et al., 2024), quantized training and inference, such as with BitNet (Ma et al., 2024), and novel approaches in fine-tuning, the use of frameworks like LangChain and AutoGen (Wu et al., 2023) and advanced prompt engineering strategies(Liu et al., 2023). These components are vital across the AI application spectrum, extending well beyond mere next token prediction. As the rankings of the competition are not based on existing ML benchmarks, we ensure the systems hold real-world value and that the winning entry in the open model track will be downloaded and used by others even after the competition is over.

Aligned with our call for transparency and reproducibility in ML, we are committed to reproducing submissions for top entries from the Open Evaluation, Open Model track. To encourage broader participation from commercial entities, the Open Evaluation, Closed Model track will not require a model release, however, we encourage participants in this track to share details such as model and infrastructure size. Competition hosts will publish our evaluation steps for both tracks to provide the community with clear, actionable guides and best practices, continuing the tradition we established in the 2023 NeurIPS Efficient Fine-tuning challenge. This approach not only democratizes access to cutting-edge research but also equips the broader community with insights for immediate application to product development in their own problem space.

1.3 Data

In collaboration with the creators of Hacker Cup, we are curating and releasing all historic questions used in Hacker Cup throughout its 12 years of history, consisting of over 100 questions. The problems, written in English, span varying levels of difficulty, with those in the final rounds being significantly more challenging than the earlier ones. The accompanying solutions for each problem are available in multiple formats, including high-level English descriptions, sketched-out algorithmic solutions, deep dives in video format with detailed explanations, and finally, code solutions in C++ and Python. This data is an unprecedented parallel coding challenge dataset across multiple modalities and programming languages. Examples of the problems and solutions are available on the Hacker Cup site. We include one past problem, solution, and details of current models' performance in the supplementary material (§4.1). Beyond our Hacker Cup dataset, participants can use any open data, for example, the competition dataset aggregated by DeepMind and released under Apache-2 license. We encourage competitors to leverage Discord communities to collaborate, curate, and share additional data. We recognize and intend to highlight that a significant amount of data has already been generated over the years across international and community-based competitive coding competitions, such as the Philadelphia Classic created by university students for Philadelphia-area high school students. Additionally, we advocate for the use of Generative AI technology to generate new synthetic data (Bauer et al., 2024), a promising trend in the recent AI research space to revolutionize data generation.

The questions used to evaluate this competition are newly written for the 2024 Hacker Cup and will only be released on the dates of the Hacker Cup rounds to all (human and AI) contestants simultaneously, thus maximizing the integrity of the competition and minimizing the potential for evaluation data leakage.

1.4 Tasks and application scenarios

The competitors will be given 4 to 7 questions to solve in each round, with points associated with each question. The The competition will have two tracks:

- Open Evaluation, Closed Model Track: Where we do not impose any limitations on resources or architecture for the system at train time, and only enforce standard competition completion times for generation.
- Open Evaluation, Open Model Track: Solutions must be built using 1 single NVIDIA A100 graphics card with 40GB VRAM, no more than 128GB RAM and no more than 1TB of storage. Similar to last year's LLM efficiency challenge, we will enforce that all solutions must be reproducible across both training and inference within 24 hours and that all solutions derive from open-sourced models, components and datasets.

1.5 Metrics

The competition comprises 1 practice round and 4 competitive rounds, each featuring 4 to 7 questions. Competitors are required to submit the output of their programs against test cases in plain text format, as specified in each problem. Each submission is scored based on the correctness of its output, compared to "gold" solutions, over a set of predefined test cases that cover various input scenarios, including edge cases.

The results will be ranked in four leaderboards: one for each of the two AI tracks, a human leaderboard for the traditional Hacker Cup competition, and a joint leaderboard comparing AI and human scores concurrently.

Ranking for AI Tracks

The ranking used to declare independent winners for each of the two AI tracks, and for awarding sponsor prizes in the Open Track, will be calculated as:

$$\text{Score} = \sum_{k=1}^{l=4} \mathbb{R}_{\mathbb{k}} \left(\sum_{j=1}^{m_k} \mathbb{w}_{jk} \right)$$

Where:

- l is a number of rounds not counting the first practice round,
- $\mathbb{R}_{\mathbb{k}}$ is the per-round scaling factor of {1, 1.25, 1.5, 1.75} to incentive higher scores for harder rounds.
- m_k is the total number of problems in round k,
- $\mathbb{w}_{jk}$ is per-problem specific number of points awarded for passing all test cases for problem j in round k.

We will announce 2 winners for this competition, one for each track above, each corresponding to one competition leaderboard.

Tie breaking for AI Tracks

If multiple competitors have the same score across rounds, we will tie break using a partial credit system based on the number of test cases that pass:

$$\text{Tie-break Score} = \sum_{k=1}^{l=4} \left(\sum_{j=1}^{m_k} \left(\max(0, \sum_{i=1}^{n_{jk}} \mathbb{1}_{ijk} - \text{Threshold}_{jk}) \right) \right)$$

Where:

- l is the number of rounds, not counting the first practice round,
- m_k is the total number of problems in round k,
- n_{jk} is the total number of test cases for problem j in round k,
- $\mathbb{1}_{ijk}$ is 1 if the test case i for problem j in round k is passed, 0 otherwise,
- Threshold$_{jk}$ is the minimum number of test cases a competitor needs to pass to receive any tie-break score for problem j in round k. This threshold is problem-dependent and will be announced after each round.

This ensures that a competitor must surpass a certain performance level to qualify for tie-breaking points, with additional points awarded based on the number of test cases correctly solved beyond this minimum requirement.

Joint Leaderboard

For informational purposes only, we will jointly rank Human and AI submissions in a single leaderboard. The scoring of AI submissions in this leaderboard will adhere to the same evaluation methodology applied to human contestants in the Hacker Cup. Specifically, credit will only be awarded if all test cases pass for a given problem:

$$\text{Ranking} = \sum_{k=1}^{l=4} \left(\sum_{j=1}^{m_k} (\mathbb{w}_{jk}) \right)$$

Where:

- l is a number of rounds not counting the first practice round,
- m_k is the total number of problems in round k,

- $\mathbb{w}_{jk}$ is a problem-specific number of points awarded if and only if all test cases pass for problem j in round k.

Finally, to foster community-building in this competition, recognition will extend beyond just the top-ranking systems in the two tracks to include a community champion chosen by vote. A voting bot will be implemented to facilitate the communication of upvotes, allowing community members to acknowledge and reward contributions that enhance the collective experience. These contributions can range from releasing base models, developing useful libraries and tools, curating and generating datasets, to publishing tutorials, fixing shared libraries, resolving build issues, and answering questions. At the competition's conclusion, the member with the highest number of upvotes in our Discord[2] server, who is neither an organizer nor an advisor of the competition, will be honored as the community champion.

1.6 Baselines, code, and material provided

General information associated with the competition will be published on our website https://llm-efficiency-challenge.github.io/

We are committed to having our updated website live shortly after the notification of competition acceptance from the NeurIPS competition committee.

We plan to provide the Hacker Cup dataset as specified in §1.3 of this proposal and a starting kit to integrate an open-source model with agent and prompt engineering support. These materials will be made available in our GitHub organization: https://github.com/llm-efficiency-challenge

The starter code and tutorials on resource-efficient fine-tuning, evaluation infrastructure and additional dataset integrated into our fork of Stanford's HELM (Liang et al., 2023)for model evaluation, which were generated for our 2023 NeurIPS competition and which will be relevant to this competition, is already located in this GitHub organization.

We plan to continue to engage community using Discord, specifically the same Discord channel we created last year and will cross-post updates to the popular CUDA MODE discord.gg/cudamode discord, a following popular Discord community built by our organizers dedicated for CUDA kernel hacking to help contestants in the open model track maximize their training and inference efficiency.

In 2023's NeurIPS competition, the organizers had engaged with community experts to build tutorials for our competition, such as this free course created with Weights and Biases and this starter guide from Lightning.ai. We plan to continue this and release a series of tutorials accessible both online and via our Discord server to cover a subset of the following topics that we deemed useful for this competition including Mixture of Experts (MoE), Model merging, Fine-tuning, Quantization for pre-training and inference, Retrieval augmented generation (RAG) with an emphasis on Graph-RAG and invited vendor talks[3] e.g: LangChain, AutoGen, Promptflow

2 Organizational aspects

2.1 Protocol

The competition adheres to the format established by Hacker Cup, consisting of 1 practice round and 4 competition rounds. Participants are required to register for a Hacker Cup account and indicate whether they are competing for the Open Model or Closed Model track. Registration is free and open to anyone except where limited by applicable laws or by Meta Hacker Cup Terms. Problems will be disclosed on the dates and times specified for the Hacker Cup competition, as detailed in section 2.3 of this document. Participants may generate code in any programming language of their choice and must submit a plain text file containing their solutions for each problem, adhering to the specified format. See the complementary material section of this proposal for an example of a submission text file format.

We will use the same infrastructure Hacker Cup uses to evaluate submissions against test cases to rank submissions. For the open model track, we will use a similar model evaluation infrastructure

[2] specific platform is subject to change

[3] specific vendors subject to change based on availability

to the one built for our 2023 NeurIPS competition to verify and reproduce the top-k results in the open model track, where $3 \leq k \leq 15$, with the actual value of k dependent on submission quality and evaluation resource constraints. Hacker Cup rules can be found in the FAQ section of the website. Our requirements for AI contestants to advance through competition rounds will mimic those for humans:

- The Practice Round may be entered by anyone. Unlike later rounds, it is open for 3 days. We highly recommend participating to gain familiarity with Hacker Cup's submission system.
- Round 1 is open to eligible registered participants.
- Round 2 is invitational, and limited to the top 5,000 placing participants in Round 1 per track, with ties broken as specified in Section 1.5.
- Round 3 is invitational, limited to the top 500 competitors from Round 2 per track.
- The Final Round (Round 4) is invitational, limited to the top 25 competitors from Round 3 per track.
- The winner of the Final Round will be the 2024 Hacker Cup champion!

2.2 Rules and Engagement

Contest rules:

The rules of this competition aim to ensure fair evaluation and reproducible results. To this end, we require the following:

1. **Open Evaluation, Open Model** track submissions must be reproducible end-to-end on a system with a NVIDIA A100 GPU with 40GB of VRAM, no more than 128GB of RAM, and no more than 1TB storage. After the competition, all teams are required to open-source their code, and methodology and link to any used data. Competition organizers will reproduce all results for the top teams before declaring a winner. Participants are required to utilize open-source code, libraries, and datasets with proper attribution or open source their own after the competition.
2. **Open Evaluation, Closed Model** track submissions will not be reproduced but must provide an API endpoint for competition hosts to query with additional validation questions, with response times required to be less than 15 minutes. Submissions in this track may optionally (and are highly encouraged to) provide details of their systems, including model and infrastructure scale.
3. Participants may compete either individually or within teams, without a limit on team size. However, joining multiple teams is strictly prohibited.
4. Organizers and advisors affiliated with the competition are prohibited from participating and claiming prizes.
5. Submissions must comply with responsible AI principles, explicitly forbidding content that could incite harm, discrimination, or engage in unethical behavior.
6. The competition operates on an honor system to foster a respectful and fair environment. Actions contrary to the spirit of the competition or otherwise unfair practices will lead to disqualification. This includes respectful communication within our Discord communities.
7. The rankings are final, and the organizers reserve the right to disqualify any participant at their discretion.
8. This competition will be governed by any additional rules as posted by the relevant Meta Hacker Cup site.

Participants are expected to adhere to these rules to maintain a fair and engaging competition for all involved. These rules cover various aspects such as reproducibility, fairness, collaboration, attribution, and the honor system. By addressing issues related to intellectual property, hardware requirements, team composition, and submission eligibility, the organizers aim to create a level playing field for participants. Moreover, the rules promote transparency and collaboration by encouraging open-source contributions and proper attribution, which can help foster a positive, sharing, innovative ML community.

2.3 Communication:

Announcements related to the competition will be made on a Discord server setup for this event https://discord.gg/XJwQ5ddMK7, as well as on our competition website https://llm-efficiency-challenge.github.io/. We encourage participants to use Discord to communicate with organizers and each other. Here, we will dedicate time to monitor and answer questions throughout the competition. While we are also reachable via DM on Discord, we prefer the use of public communication channels for technical questions, as this allows the answers to benefit the entire community. Additionally, participants are welcome to create GitHub issues for detailed discussions.

2.4 Schedule and readiness

Detailed timeline

The following timeline is tailored to align with the Hacker Cup and NeurIPS schedules, anticipating the official schedule announcement of the 2024 cycle in early July 2024. For reference, you can view the 2023 Hacker Cup schedule here.

We acknowledge that our operational dates extend beyond October 2024, however, giving the challenging nature of the problems, we expect the top contenders to emerge by round two, at the end of October; and we can start vetting their solutions. If a small group of competitors does indeed make breakthroughs, we would continue scoring them until the end of Round 4 in December. This will only incur a small logistical overhead and is unlikely to impact our operations.

- End of May 2024: Announcement for AI track in Hacker Cup 2024, release starter-kit and Hacker Cup dataset
- Early July 2024: Hacker Cup 2024 schedule to be released, finalize Hacker Cup questions.
- July-Aug 2024: Tutorials and lectures on Discord
- Late September 2024: Practice Round
- Early October 2024: Round 1
- End of October 2024: Round 2, Notify top k-teams
- Early November 2024: Round 3
- Mid-November 2024: Vet top-k solution in open track.
- Early December 2024: Final round, announce winners.
- December 8th - 12th 2024: in-person workshop
- February 29th, 2025: Draft of competition paper
- March 2025: Next iteration of the competition proposal
- May 2025: Submit competition retrospective paper to NeurIPS 2025 D&B track.

2.5 Competition promotion and incentives

Promotion: We intend to leverage the established Hacker Cup presence, along with the institutional social media accounts of organizers, such as Microsoft Research's official X and LinkedIn accounts to promote this competition. Additionally, the organizers have previously achieved notable success in establishing a social media presence and cultivating communities. Our 2023 NeurIPS efficiency fine-tuning competition was highlighted at the top of HackerNews Thread 1 Thread 2, and we successfully developed communities for that competition on Discord with 1,400+ members and 186 registered teams. At the NeurIPS conference, the 2023 efficiency challenge had the largest number of pre-registrants among all competitions, topping over 600 individuals. Our second iteration is a first in establishing an AI track in a major competitive programming competition, we therefore expect significantly more media coverage for this event.

Emphasizing the importance of cultivating a diverse and inclusive ML community is a priority for the organizers. We plan to engage with organizations such as Black in AI and the Grace Hopper Celebration community, encouraging their members to participate in our competition. This year we are placing special emphasis on community building and engaging the next generation, as we observed some of the most active participants from our 2023 competitions were high school students.

Recognizing the strong appeal of competitive programming to this demographic, we plan to further cultivate this trend by reaching out to some of the STEM youth communities, such as DigiGirlz to provide tutorials accessible to their level, further nurturing their interest and involvement of a younger generation.

Incentives: Through Hacker Cup, human participants have access to tiered, cash prizes ranging from $200 to $20,000 based on placement. We expect similar levels of prizes for our Open Model track, with specific cash values TBD pending ongoing sponsorship discussions. We do not expect to provide cash prizes for the Closed Model track.

Last year, we had great success in securing community-based incentives and rewards for our efficiency competition. Organizers obtained sponsorship for over $30,000 in prizes from sponsors such as Microsoft Research and Mozilla.ai, and distributed prizes among both top overall winners and top students. Additionally, we secured compute grants from AWS and Lambda Labs for our 2023 competition. We plan to continue this strategy, facilitating connections between recipients and donors, and awarding top participants in the Open track.

Lastly, we will provide one NeurIPS 2024 in-person conference pass to our community champion and another one to our Open Model track winner. Additionally, we anticipate inviting participants with innovative solutions to be co-authors of our NeurIPS 2025 paper.

3 Resources

3.1 Resources provided by organizers

The organizers plan to dedicate over 1,500 hours of volunteer effort to guarantee the success of this event, encompassing tasks such as curating data, establishing the codebase, crafting tutorials, answering queries, fostering community engagement, and securing sponsorships. Thus far, we've successfully partnered with Meta's Hacker Cup to host the event and access to their historical data and secured over 1,000 hours of GPU compute time.

To illustrate the level of commitment from the organizers for our 2023 NeurIPS competition, we collectively achieved significant milestones, and we anticipate dedicating even more effort towards this year's competition:

- A brand new Discord community with 1,400+ people
- Two evaluation systems, a Discord bot for inference and another for training reproduction.
- 700+ successful submissions on our Discord Leaderboard
- 186 registered teams and 225 final submission
- 800+ volunteer hours and 1000+ GPU hours for the evaluation alone
- Organized talks from creators Phi and Qwen models, PEFT library, Flash-HELM for our NeurIPS 2023 in-person session

Lastly, we would like to note that many of our employers offer generous employee volunteer matching programs, meaning some of the hours we spend volunteering will be matched with donations to the NeurIPS organization.

3.2 Support requested

We kindly request NeurIPS' support in helping non-US teams secure necessary travel documents, enabling them to participate in the in-person event. We also request assistance in organizing the in-person event, including providing facilities, staff, AV support, and guidance to ensure a successful competition. Additionally, any further support from NeurIPS, such as promoting the competition through their official website, connecting us with relevant experts, or offering logistical assistance, would be greatly beneficial to our efforts to democratize LLM development and make it more accessible to a broader audience.

3.3 Organizing team

Through our 2023 NeurIPS LLM efficiency fine-tuning competition, we built a close-knit community among the organizers, and we are thrilled to have most of the team returning for another round this year, along with some new additions.

Weiwei Yang is a Research Director leading an applied machine learning team at Microsoft Research (MSR). Her research interests lie in resource-efficient learning methods inspired by biological learning. Weiwei aims to democratize AI by addressing sustainability, robustness, scalability, and efficiency in ML. She has successfully applied her research to organizational science, countering human trafficking, and stabilizing energy grids. Before joining Microsoft Research, Weiwei worked extensively in Bay Area startups and managed several engineering teams. She has experience as both an organizer and key participant in numerous workshops and research events hosted by Microsoft Research, and the United Nations, and served as the co-lead organizer for the NeurIPS 2021 Out of Domain Adaptation and Generalization workshop and 2023 NeurIPS Large Language Model Efficiency Challenge. Weiwei will serve as the co-lead organizer for this competition, coordinating organizers, interfacing with academic and corporate sponsors and engaging with the broader ML community

Mark Saroufim Mark is a PyTorch engineer focused on model inference, compilers and Generative AI performance. Broadly, Mark cares about 2 things making ML models faster and helping others make their ML models faster. Mark has been instrumental in establishing and growing several influential communities, including Breaking Stagnation, The Robot Overlord Manual CUDA MODE and the NeurIPS 2023 LLM Efficiency group. These communities serve as platforms for researchers, developers, and enthusiasts to collaborate and drive advancements in machine learning. Mark graduated from UC San Diego with a focus on Machine Learning Theory. Mark will serve as the co-lead organizer for this competition, coordinating organizers, interfacing with academic and corporate consultants and sponsors, as well as engaging with the broader ML community

Joe Isaacson is an engineering manager at Meta working on PyTorch. His interests span natural language understanding, recommendation systems at scale, and machine learning for biological applications. Before Meta, Joe led machine learning groups at Asimov, a synthetic biology company based in Boston, and Quora, wherein he helped to launch the Quora Question Pairs Kaggle Competition. Joe will serve as the co-lead organizer, assisting with the logistics of running this competition. His responsibilities will include interfacing with sponsors and coordinating with the engineering and machine learning communities to ensure a successful event.

Luca Antiga Luca Antiga is the CTO at Lightning AI. He is an early contributor to PyTorch core and co-authored "Deep Learning with PyTorch" (published by Manning). He started his journey as a researcher in Bioengineering and later co-founded Orobix, a company focused on building and deploying AI in production settings.

Greg Bowyer Greg Bowyer is the chief engineer at TicketMaster and a long-term open-source contributor and ML enthusiast. He built the training evaluation framework for our 2023 NeurIPS Large Language Model Efficiency Challenge, and will be helping with training evaluation infrastructure this year.

Driss Guessous is a machine learning engineer at Meta working on the PyTorch Core library. His primary work on PyTorch has been to develop efficient and composable building blocks for transformer-based models. Before joining Meta, Driss worked as a machine learning engineer for Funnel Leasing building NLP-based product solutions. He received a MCS from the University of Illinois at Urbana-Champaign and a dual bachelor's degree in Physics and Applied Mathematics from The Ohio State University. Driss will be responsible for establishing the evaluation pipeline, answering technical inquiries from the community, and reproducing the winning models.

Christian Puhrsch is a software engineer at Meta working on PyTorch. These days he is primarily focused on sparsity, quantization, and efficient representations of ragged data for deep learning with applications in NLP, Vision, and Audio. Prior to joining Meta in August 2015, Christian finished his Master of Data Science at NYU.

Supriya Rao is an engineering manager at Meta working on PyTorch. Her interests lie in accelerating AI models using innovative model optimization techniques and making AI more accessible to the general community. Prior to Meta, she worked at Nvidia, on optimizing memory subsystems for next gen GPUs and TensorRT for efficient inference on GPUs.

Geeta Chauhan leads the Applied AI group at Meta working on strategic initiatives for PyTorch for taking research to production. Her interests include Sustainable AI, scaling for large foundation model training, model optimization, and inference. She is a winner of Women in IT – Silicon Valley – CTO of the Year 2019, an ACM Distinguished Speaker, and a thought leader on topics ranging from Sustainability, Responsible AI, LLMs, and Deep Learning. She is passionate about promoting the use of AI for Good and having a positive impact on climate change.

Margaret Li is a PhD student at the University of Washington, advised by Luke Zettlemoyer and Tim Althoff, and is concurrently a visiting researcher at FAIR (Meta). Her research interests are primarily in efficient pre-training and continued training of large language models through specialization, sparsity, and compositionality, as well as post-training merging and re-use of models. She was previously an organizer and problem writer for the Philadelphia Classic.

David Harmeyer is a software engineer at Meta working in Recruiting Products. He's a Meta Hacker Cup Judge who has authored over 100 programming contest problems. He competed in ICPC World Finals twice representing the University of Central Florida, finishing in 17th place in Moscow. He currently competes as SecondThread on Codeforces. At peak rating in 2021, he was a Codeforces International Grandmaster, reaching the 7th highest rating in the United States. He now runs the competitive programming YouTube channel SecondThread.

Wesley May is a software engineer at Meta working in Recruiting Products. He's a Meta Hacker Cup Judge and is a specialist in natural language processing, algorithm design, game design, Java, Python, C++, and contract bridge.

4 Supplementary Material

4.1 Sample Hacker Cup problem

Nim Sum Dim Sum, a bustling local dumpling restaurant, has two game theory-loving servers named, you guessed it, Alice and Bob. Its dining area can be represented as a two-dimensional grid of R rows (numbered $1..R$ from top to bottom) by C columns (numbered $1..C$ from left to right).

Currently, both of them are standing at coordinates $(1, 1)$ where there is a big cart of dim sum. Their job is to work together to push the cart to a customer at coordinates (R, C). To make the job more interesting, they've turned it into a game.

Alice and Bob will take turns pushing the cart. On Alice's turn, the cart must be moved between 1 and A units down. On Bob's turn, the cart must be moved between 1 and B units to the right. The cart may not be moved out of the grid. If the cart is already at row R on Alice's turn or column C on Bob's turn, then that person loses their turn.

The "winner" is the person to ultimately move the cart to (R, C) and thus get all the recognition from the customer. Alice pushes first. Does she have a guaranteed winning strategy?

Constraints

$1 \leq T \leq 500$

$2 \leq R, C \leq 10^9$

$1 \leq A < R$

$1 \leq B < C$

Input Format

Input begins with an integer T, the number of test cases. Each case will contain one line with four space-separated integers, R, C, A, and B.

Output Format

For the ith test case, print "Case #i: " followed by "YES" if Alice has a guaranteed winning strategy, or "NO" otherwise.

Sample Explanation

The first case is depicted below, with Alice's moves in red and Bob's in blue. Alice moves down, and Bob moves right to win immediately. There is no other valid sequence of moves, so Alice has no guaranteed winning strategy.

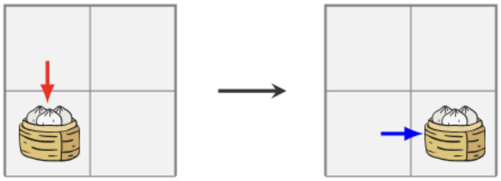

Figure 1: First case illustration

The second case is depicted below. One possible guaranteed winning strategy is if Alice moves 3 units down, then Bob can only move 1 unit, and finally Alice can win with 1 unit.

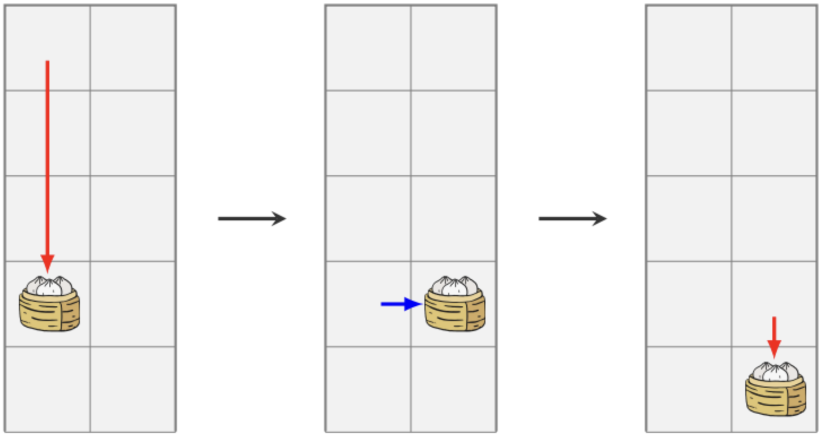

Figure 2: Second case illustration

Sample Input

```
3
2 2 1 1
5 2 3 1
4 4 3 3
```

Sample Output

```
Case #1: NO
Case #2: YES
Case #3: NO
```

Solution

If Alice reaches row R before Bob reaches row C, then it's game over for Alice. Since each player now wants to get to the finish as slowly as possible, both have a simple dominating strategy of only moving 1 unit in their direction each turn, and R and C are the only things that matter.

If $R \leq C$, Bob can always force Alice to reach row R first by moving 1 unit right at a time. Alice also only moves 1 unit at a time, because if she moves any faster, she'll just get stuck sooner.

Conversely, if $R > C$, then Alice can always force a win by moving 1 step at a time. Therefore we output "YES" if and only if $R > C$, regardless of the values of A and B.

Existing model performance

In preliminary explorations, we tested the ability of existing state-of-the-art models to solve this problem, along with others from the 2023 Hacker Cup. Specifically, we experimented with various zero-shot Chain-of-Thought (COT) prompts (e.g., "Let's think step by step:..."), and prompt-chaining with various breakdowns of problem solving steps (e.g., "Brainstorm some ideas...", "Identify some key insights...", "Write a high level sketch of a solution..."), written in a general way, without including the details of any problem. We also tried to incorporate knowledge of the problems in our prompt-chains (e.g. "Alice should not take steps of size A. What should she do instead?"), as if a knowledgeable human were giving the model hints to the solution. For these explorations, we used GPT-4 and a further fine-tuned Code Llama 70B.

Across our 50+ attempts with these models, no model was able to successfully solve this problem. The models were usually able to make several correct though largely superficial observations about the game like "the game has a terminal state." However, they would also include incorrect reasoning or leave out crucial insights, and ultimately did not succeed in generating a fully correct explanation of a solution or a code solution. There were a few common failure points: (1) the model would identify this as 2-D Nim despite major differences in rules, and would print a sometimes-valid solution to 2-D Nim, (2) the model would assert that both players needed to take steps of the maximum size, A and B for Alice and Bob, respectively, (3), the model explicitly or implicitly states that the game is symmetric for Alice and Bob, or makes related assumptions, such as that $R = C$, when proposing a winning strategy for Alice, (4) the model would output a dynamic programming or recursion based solution that may sometimes be technically correct without constraints, but would fail to complete within the test case constraints given in the problem, as a direct result of excessive runtime complexity at $O(ABRC)$, as opposed to the optimal solution's $O(1)$ runtime.

A few other observations of the models' behavior during these trials, as well as when tested on other 2023 Hacker Cup Practice Round questions include: Models are highly likely to generate code even when asked not to (e.g. "Brainstorm ideas for solutions, but do not generate any code..."). When models generate an explanation of or commentary on their code, followed by code, the two do not always correspond, though they usually seem to on first glance. When models were asked to critique their own work ("Is this correct?" or "Does your code correctly pass this test case?"), models would sometimes refuse to answer, or assert that their solution was correct and re-state the justification. When models stated that their solutions were incorrect, however, they would frequently offer to fix the solution but generate the same code verbatim, though there were also occasions when the model would fix one oversight and still fail to generate a correct solution.

Given that this is a practice round question, and that current commonly-used state-of-the-art level coding-specific LLMs consistently fail, we believe that there is significant room for models to improve.

References

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes, 2024.

André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. Comprehensive exploration of synthetic data generation: A survey, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. Science, 378(6624):1092–1097, December 2022. ISSN 1095-9203. doi: 10.1126/science.abq1158. URL `http://dx.doi.org/10.1126/science.abq1158`.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Comput. Surv., 55(9), jan 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL `https://doi.org/10.1145/3560815`.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik

Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with alphacodium: From prompt engineering to flow engineering, 2024.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.

Yue Wang, Weishi Wang, Shafiq R. Joty, and Steven C. H. Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In EMNLP, pp. 8696–8708. Association for Computational Linguistics, 2021.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang (Eric) Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, August 2023.