SGU-Editorial: A Small Dataset of Competitive Programming Problems with LLM-Enhanced Editorials

Radoslav Dimitrov
contact@radoslav11.com

Abstract

State-of-the-art large language models (LLMs) perform poorly on problems from the Saratov State University (SGU) Online Judge, likely because SGU problems feature a diverse range of algorithmic ideas underrepresented in existing training datasets. To address this, we present SGU-Editorial, a small dataset containing 250 competitive programming problems enhanced with detailed editorials. In this work we present v1, the first version of the dataset. Each problem includes the original statement, accepted solutions in C++ or Python, and LLM-enhanced editorials containing algorithm explanations, implementation guides, complexity analysis, and reference implementations. Future versions will expand the problem coverage as we continue solving the SGU archive.
Keywords: competitive programming, dataset, code reasoning, LLM, editorial generation, algorithm explanation

1 Introduction

The intersection of large language models (LLMs) and competitive programming has emerged as an important area of research. Recent advances have shown that LLMs can solve programming problems at varying difficulty levels, from basic coding exercises to International Olympiad in Informatics (IOI) problems [1, 5]. For example, OlympicCoder models have demonstrated strong performance on IOI problems, even outperforming some closed-source models [7]. However, state-of-the-art models still struggle with certain problem sources. In particular, we observe that LLMs perform poorly on problems from the Saratov State University (SGU) Online Judge.

The SGU (Saratov State University) Online Judge, known as acm.sgu.ru, was created approximately 20 years ago by competitive programming enthusiasts from Saratov University, including Mike Mirzayanov (who later founded Codeforces) and Andrew Lazarev. The platform hosted ACM ICPC-style problems and was widely used for training. Although the original infrastructure became outdated, the problems were preserved and migrated to Codeforces as a dedicated "acmsguru" section [4], reflecting continued interest in this historical problemset.

We hypothesize that LLMs struggle with SGU problems because they feature a diverse range of algorithmic ideas and problem-solving techniques that are underrepresented in existing competitive programming datasets. The SGU archive contains problems with unique formulations and solution approaches that may not appear frequently in more commonly scraped sources like modern Codeforces contests or LeetCode.

Existing competitive programming datasets such as POJ-104 [6], COFO [2], and CodeContests [5] focus primarily on collecting source code submissions for tasks like program classification, clone detection, and code generation. While valuable, these datasets draw from a limited set of sources and typically lack detailed explanations of *why* particular algorithms work, *how* to arrive at the solution, and *what* insights are needed to solve the problem.

In competitive programming, such explanations are called *editorials*—documents that provide problem analysis, algorithm descriptions, implementation guidance, and complexity analysis. Editorials are essential learning resources but are time-consuming to write and often unavailable or incomplete for many problems.

In this work, we present SGU-Editorial, a small dataset that addresses this gap. We present v1, the first version of the dataset. It contains:

1. A curated collection of 250 competitive programming problems from the SGU Online Judge with original problem statements and *commented* accepted solutions. Future versions will expand the coverage.
2. LLM-generated editorials for each problem, created using reasoning models (o1, o3 and GPT-5 series), containing structured explanations with algorithm analysis, implementation guides, and reference solutions in both C++ and Python.

We release this initial version publicly to enable early research while we continue expanding the dataset.

2 Related Work

Several datasets have been proposed for competitive programming and code understanding tasks.

POJ-104 [6] consists of 104 programming problems from the Peking University Online Judge with approximately 500 C/C++ solutions per problem, totaling around 52K programs. It is primarily used for program classification tasks.

COFO [2] is a larger dataset scraped from Codeforces containing 809 problems with 369K source codes in C, C++, Java, and Python. It includes metadata such as problem tags, specifications, and test cases, targeting classification and tagging tasks.

CodeContests [5] is a dataset of competitive programming problems used to train AlphaCode, containing problems from Codeforces, Description2Code, and CodeNet with input-output examples and solutions.

APPS [3] contains 10,000 coding problems of varying difficulty with test cases and solutions, designed for evaluating code generation capabilities.

These datasets focus on the *code* aspect of competitive programming. In contrast, SGU-Editorial focuses on the *reasoning* aspect by providing detailed editorials that explain the problem-solving process.

3 Methodology

3.1 Source Problems

The SGU (Saratov State University) Online Judge was one of the pioneering competitive programming platforms, hosting problems from ACM ICPC-style competitions. The problems span various algorithmic topics including graph theory, dynamic programming, number theory, computational geometry, and combinatorics.

We collected 250 problems from the SGU archive. For each problem, we obtained:

- The original problem statement with input/output specifications.
- Sample input and output test cases.
- An accepted solution written by the first author in C++ or Python.

3.2 Editorial Generation

For each problem, we generated editorials using a combination of reasoning-capable language models, including OpenAI's o1, o3, GPT-5, GPT-5.1, and GPT-5.2. The input to each model consisted of the problem statement, sample I/O, and an accepted solution. The generated editorials were then manually reviewed and edited to correct any inaccuracies or improve clarity.

3.3 Quality Assurance

All solutions in the dataset are *accepted* solutions that have passed the online judge's test suite. Furthermore, a large chunk of the solutions were made to already contain a brief description of the approach as comments. This makes LLM-generated editorials on average being in a good shape, but we also did do some basic spot-checking for correctness.

4 Data Description

The dataset is organized in two main directories: `dataset/` and `problems/`. The `dataset/` directory contains three files for each problem: `pXXX.txt`, the enhanced editorial; `pXXX_raw.txt`, which contains the original problem statement, solution, and sample I/O; and `pXXX_finetune.txt`, which is a version of the data formatted for model training. The `problems/` directory contains the same information as the `_raw.txt` files, but organized as individual files for each problem (e.g., `statement.txt`, `pXXX.cpp`).

Each editorial follows a consistent structure, including a concise problem statement, a detailed editorial with algorithm explanations, commented reference implementations in C++ and Python, and a brief summary for experienced programmers.

The dataset contains 250 problems, with 227 solutions in C++ and 27 in Python. The average editorial length is approximately 1317 tokens, and the average solution length is approximately 130 lines of code.

5 Future Work

SGU-Editorial is an ongoing effort. We plan to extend this work in several directions:

Expanded Coverage: We are actively solving additional SGU problems. Future versions will include more problems as we work through the archive, eventually aiming for comprehensive coverage.

Finetuning Analysis: We intend to conduct experiments finetuning LLMs on the SGU-Editorial dataset to measure whether exposure to these problems and editorials improves model performance on SGU and similar challenging problem sets. Some preliminary experiments with small open source models suggest that this might be the case.

Evaluation Benchmark: We plan to develop SGU-Editorial into a proper evaluation benchmark by generating comprehensive test data for each problem. This would enable standardized evaluation of code generation models on problems that current state-of-the-art systems find difficult.

6 Conclusion

We presented SGU-Editorial, a small dataset containing 250 competitive programming problems from the SGU Online Judge, enhanced with detailed LLM-generated editorials. In this work we presented v1, the first version of the dataset. SGU problems present a challenge for current state-of-the-art LLMs, likely due to the diverse algorithmic ideas underrepresented in existing training data. Each problem in our dataset includes a structured editorial with algorithm explanations, implementation guides, and reference solutions in multiple languages.

SGU-Editorial has certain limitations, including its relatively small scale (250 problems), the provision of a single reference solution per problem, and the potential for subtle inaccuracies in LLM-generated editorials.

Data Availability

The dataset is continuously updated and available at `https://github.com/radoslav11/acm-sgu`. The complete dataset for v1, which is presented in this work, is available as a single zip archive at `https://radoslav11.com/sgu-dataset/sgu-editorial-dataset-v1.zip`.

Acknowledgments

We thank Saratov State University for the original problem archive, and OpenAI for access to the reasoning models used in editorial generation.

References

[1] Google DeepMind AlphaCode Team. Alphacode 2 technical report. 2023.

[2] Kuldeep Gautam, S VenkataKeerthy, and Ramakrishna Upadrasta. Cofo: Codeforces dataset for program classification, recognition and tagging. *arXiv preprint arXiv:2503.18251*, 2025.

[3] Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021.

[4] Vitaliy Kudasov. Codeforces: acm.sgu.ru comes back. `https://codeforces.com/blog/entry/59070`, 2018. Codeforces Blog.

[5] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

[6] Lili Mou, Ge Li, Lu Zhang, Tao Wang, and Zhi Jin. Convolutional neural networks over tree structures for programming language processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[7] Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. Codeforces cots. `https://huggingface.co/datasets/open-r1/codeforces-cots`, 2025.