Introduction

- The major challenges in solving competitive-level programming problems pertain more to reasoning than to solution implementation.
- 2. LLMs lack the ability to plan ahead, so once it made a mistake in its early stage of reasoning, it will persist with it, which is hard to fix through debugging.
- Data of commented/explained solutions is inadequate and expensive to annotate. Data available is <problem, solution> pairs. *Solution = Program/Implementation*

**Framework:**

- The (structured) explanation generation and evaluation framework and corresponding prompts (Top). An example of the full explain prompt (Bottom Left).
- Solve@10 from 6.1% to 9.1% on GPT-turbo-3.5 (Solve@k; solve rate when generating k candidate programs)

**Can Chain-of-Thought help?**

**Explain the Solution**

- Though solve-rate is poor, reasoning process is also poor, but given the golden solutions, LLMs are better at explaining them. <Problem, Solution> ➡<Problem, Explanation, Solution>

**Problem**: simplified: Given an array of n integers, you are allowed to swap the signs (positive or negative) between any pair of integers for a range (arr1, arr2).

```python
def solve(arr):
    h = int(arr[i] if arr[i] else 0)
    for i in range(len(arr[0:arr1[0]])): (abs(arr[i]) <= 1)
    print("yes" if 1 in arr[::-1] else "no")
```

**Explanation:** Swapping signs can be seen as moving signs arbitrarily. A non-decreasing array must have negative elements ahead of positive ones and moving negative signs ahead is the optimal operation can be made.

**Dataset**

- Dataset Source: Codeforces problems after Sep 2021, ensuring GPT-3.5/GPT-4 **hasn’t seen** the problems.
- CodeContests (Li et al., 2022): 165 problems with rankings (difficulty level) from 800 to 3600.
- Our Data: 50 problems with rankings from 800 to 2000.

**Human Evaluation**

- We use Human Author Likert scores (~2: very poor to 2: excellent) to evaluate various aspects of the explanations.

**Experiments**

- Both GPT-3.5 and GPT-4 are good at describing the solution, while GPT-4 is much better in explaining it clear with no ambiguity and showing understanding of the key idea behind.

**Automatic Evaluation**

- We further evaluate the usefulness: How much can the silver explanation aid the problem solving?

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Solve</th>
<th>Wrong Answer</th>
<th>BLEU</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.1%</td>
<td>15.0%</td>
<td>40.0%</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>UsRlug</td>
<td>30.1%</td>
<td>18.9%</td>
<td>42.1%</td>
<td>0%</td>
</tr>
<tr>
<td>Silver</td>
<td>75.6%</td>
<td>11.4%</td>
<td>11.4%</td>
<td>1.6%</td>
</tr>
<tr>
<td>OneStern</td>
<td>50.0%</td>
<td>27.0%</td>
<td>16.0%</td>
<td>1.5%</td>
</tr>
</tbody>
</table>

**Table 5:** Final judgement of generated programs that pass the public tests. BLEU mean time limit exceeded, and other includes memory limit exceeded and runtime error.

**Conclusion and Future Work**

- We propose to use LLMs to generate structured explanations given <problem, solution> for competitive programming solutions. Experiments show that the explanations can 1) satisfying the human programming expert who authored the oracle solution, and 2) aiding LLMs in solving problems more effectively.

- If generated at scale, can silver explanations be used as a source to improve subsequent problem-solving?