Final Project (Part 2): Leveraging Generative AI for Debugging Complex DP Solutions

Deadline: 23:59, May 22, 2024 (Wed.)
Weight: 8% of your total course grade.

In Part 1 of your course project, you tackled a challenging dynamic programming optimization problem involving multiple complex constraints, including directional movement limits, negative reward cells, and gas consumption. Despite the high complexity, 90% of you successfully devised workable DP solutions. Part 1 explicitly prohibited the use of generative AI tools to ensure students' individual DP proficiency.

In Part 2, we explore leveraging generative AI as a debugging assistant. You will be provided with two Python implementations, each containing deliberate bugs. You can choose to use generative AI (e.g., ChatGPT, GitHub Copilot, or similar tools) to help debug one or both provided codes. Your goal is to submit a fully debugged, working version of at least one code snippet, along with your interaction history with the AI assistant.

1. Project Description

We provide two Python implementations:

Easier-to-debug version: This version (i.e., easy function in solution.py) contains relatively simple bugs suitable for effective debugging with generative AI.

Harder-to-debug version: This version (i.e., hard function in solution.py) includes subtle and challenging bugs, making AI often fails to fix the errors.

You are required to use AI to debug both versions and correct the bugs in at least one of them. If you successfully fix the bugs in both, you can obtain bonus points (10%).

Both functions are provided in "*solution.py*", and you are not allowed to change function names. You can evaluate function correctness by execute *"python test.py hard"* or *"python test.py easy".*

2. Submission Requirements

Your submission must clearly demonstrate your debugging process, AI interactions, and learning outcomes. Specifically, submit the followings:

1) Debugged Python Code:

Submit a clearly commented and fully functional Python implementation of at least one of the provided code snippets.

2) Generative AI Interaction History:

Include your complete interaction history with the AI tool.

3) Written Report (400 words):

(a) Clearly explain each bug identified with the help of generative AI. You may directly use AI-generated explanations or rewrite them in your own words. Note: Generative AI has hallucinations, and bugs reported by it might not always be correct.

(b) Provide a summary of your observations during the use of generative AI, including: Which generative AI model you found most effective; Instance of prompts that were particularly useful; Common issues or cases where generative AI frequently failed to provide helpful assistance, etc.

3. Grading Criteria

Item	Description	Weight
Code Correctness	Accurate, fully functioning debugged code.	40%
Bugs Explanation	Part a of the written report	30%
Chat Log	Save your chat session as a PDF file.	20%
Reflection	Part b of the written report	10%

4. Bonus

Successfully debugging **both** code snippets with documented AI assistance earns an additional 10% bonus.

Academic Integrity Note

Use of generative AI tools is explicitly allowed.