Using ChatGPT to Solve LeetCode Problems Efficiently

Step 1: Understand the Problem

- "Can you explain this LeetCode problem in simple terms?"
- "What is this problem really asking?"
- "What are the key constraints and edge cases I should look out for?"
- "Can you give me a small dry-run example of how this works?"

Step 2: Plan the Approach (before coding)

- "What kind of approach would you try for this problem?"
- "Which data structures or algorithms might be useful here?"
- "What are some common patterns that apply to this type of problem?"
- "Can you give me a hint, not the full code?"

Step 3: Debug or Unblock

- "Here's my code. It's failing test case X. Can you help me find the bug?"
- "Why is this solution getting Time Limit Exceeded (TLE)?"
- "How can I optimize this part of the code?"

Step 4: Clean Up & Optimize

- "Can you review this code and suggest improvements?"
- "Can this be done more efficiently?"
- "What's the time and space complexity of this solution?"
- "Is there a more Pythonic/cleaner way to write this?"

Step 5: Learn the Pattern

- "What algorithmic pattern does this problem follow?"

- "What are 5 similar LeetCode problems based on the same idea?"

- "How would you generalize this solution for other problems?"

Bonus: Simulate a Real Interview

- "Act like an interviewer. Give me a Medium LeetCode-style problem."

- "Wait for me to explain my thought process step by step."

- "Only give me hints if I'm stuck for more than 3 minutes."

Example Full Flow

1. "Here's the LeetCode problem: [paste link or description]. Can you explain it simply?"

2. "Now I want to solve it myself. Just give me a high-level hint."

3. [You try it]

4. "Here's my attempt. It's failing on some input. Can you help me debug?"

5. "Cool, it's working now. How can I improve this solution? Any cleaner way?"

6. "What pattern is this problem using? Can you give me similar problems?"

7. "Give me a LeetCode-style question to test myself on this pattern."