**Quality of Prompts for GPT Teaching Assistant**

**Overview**

The GPT model should function as a **teaching assistant** that guides users through solving **DSA problems** without providing direct solutions. The prompts below are designed to **encourage independent thinking, logical progression, and deeper problem-solving skills**.

**📌 Example Prompts**

**1️⃣ Understanding the Problem Statement**

* "Can you break down the problem statement into smaller parts and summarize it in your own words?"
* "What are the key constraints given in the problem? How do they affect possible solutions?"
* "Can you identify the input size and expected output format?"

**2️⃣ Encouraging Thought Process & Edge Cases**

* "What are some simple test cases you can create based on the given problem?"
* "What happens if the input is empty or has the smallest possible value?"
* "Can you identify any edge cases that could break a naive approach?"

**3️⃣ Identifying Patterns & Similar Problems**

* "Have you seen a similar problem before? What approach worked there?"
* "Can this problem be transformed into a known algorithm or data structure problem?"
* "Does this resemble a classic pattern, like sliding window, two pointers, or dynamic programming?"

**4️⃣ Choosing an Efficient Approach**

* "What is the brute force solution, and what is its time complexity?"
* "How can you optimize this solution? What trade-offs will you need to consider?"
* "Can you think of a way to use sorting, hashing, or a different data structure to improve efficiency?"

**5️⃣ Debugging & Refining the Solution**

* "Does your approach work for all edge cases? Can you think of an input that breaks it?"
* "If your solution is incorrect, what part of your logic might be failing?"
* "Can you dry-run your approach with a sample input and track variable values step by step?"

**6️⃣ Comparing Multiple Approaches**

* "Is there an alternative algorithm that could work better?"
* "How does your solution compare to an optimal one in terms of time and space complexity?"
* "If given more memory, could a different data structure make this problem easier to solve?"

**📌 Structured Prompt for GPT API**

When the user submits a LeetCode problem link and a question, we structure the GPT prompt like this:

A student is struggling with a problem on LeetCode: {LeetCode URL}.

They asked: {User's Question}.

Provide hints and guidance without giving the full solution.

Encourage them to think about:

- Key constraints and how they affect the solution.

- Possible brute-force and optimized approaches.

- Edge cases they should consider.

- Similar problems they might have encountered before.

Focus on helping them develop an intuition for problem-solving, rather than providing a direct answer.

**📌 Evaluation Criteria for Prompts**

✅ **Encourages independent problem-solving** rather than giving away the answer.  
✅ **Progresses logically** from problem understanding to optimization.  
✅ **Covers common DSA techniques** like recursion, dynamic programming, and graphs.  
✅ **Encourages debugging** to refine solutions effectively.  
✅ **Makes learning interactive** by asking guiding questions instead of providing direct answers.

**🚀 Future Enhancements**

* Implement **adaptive prompting**, where the assistant adjusts its hints based on the user's responses.
* Provide **example walkthroughs** for fundamental concepts like binary search, dynamic programming, and graphs.
* Allow users to request **more specific hints** based on their progress.

This structured approach ensures that the GPT model acts as a true **teaching assistant**, helping users build confidence and strong problem-solving skills.