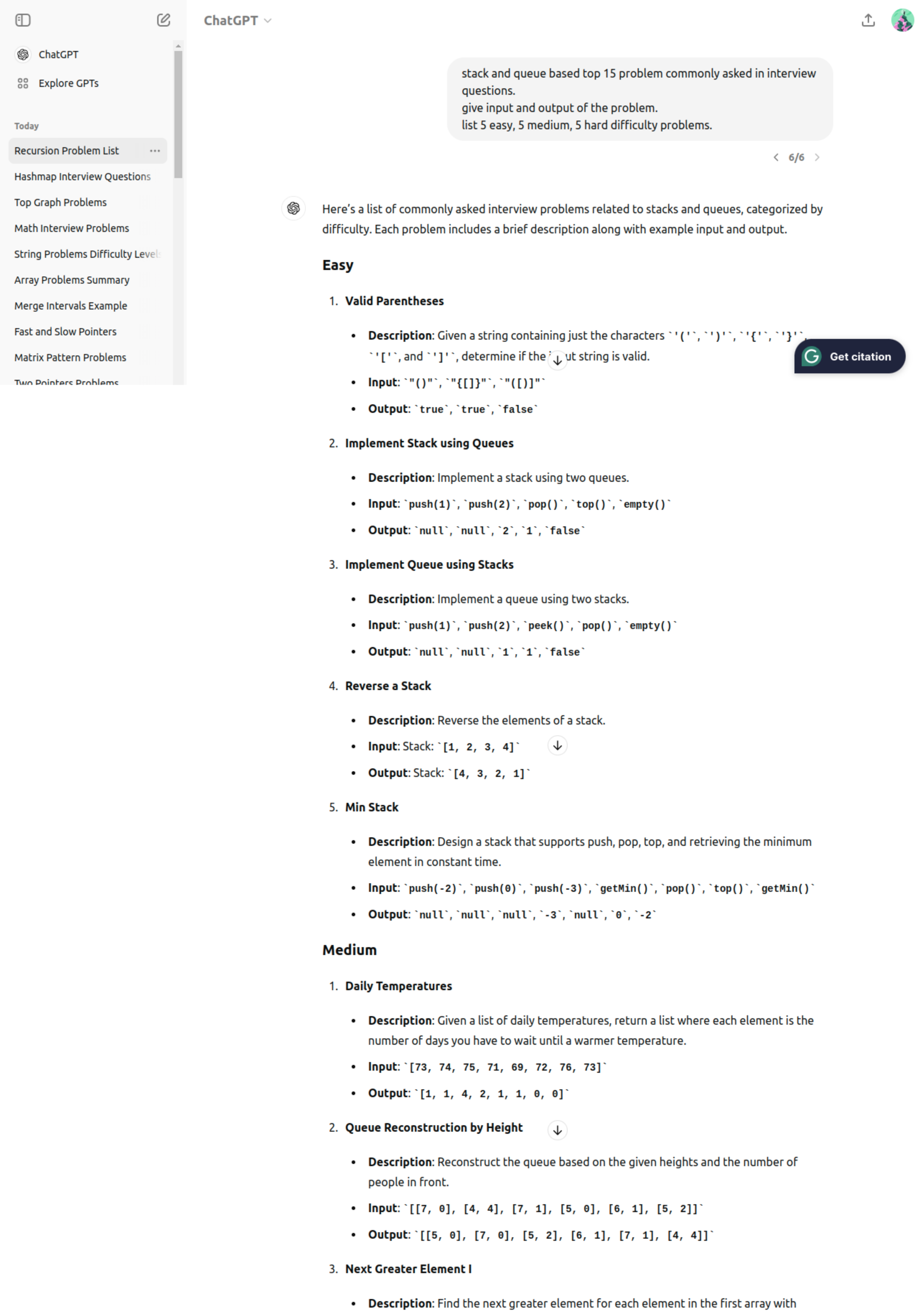

stack and queue based top 15 problem commonly asked in interview questions.
give input and output of the problem.
list 5 easy, 5 medium, 5 hard difficulty problems.

Here's a list of commonly asked interview problems related to stacks and queues, categorized by difficulty. Each problem includes a brief description along with example input and output.

Easy

1. **Valid Parentheses**
   - **Description**: Given a string containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['`, and `']'`, determine if the input string is valid.
   - **Input**: `"()"`, `"{[]}"`, `"([)]"`
   - **Output**: `true`, `true`, `false`
2. **Implement Stack using Queues**
   - **Description**: Implement a stack using two queues.
   - **Input**: `push(1)`, `push(2)`, `pop()`, `top()`, `empty()`
   - **Output**: `null`, `null`, `2`, `1`, `false`
3. **Implement Queue using Stacks**
   - **Description**: Implement a queue using two stacks.
   - **Input**: `push(1)`, `push(2)`, `peek()`, `pop()`, `empty()`
   - **Output**: `null`, `null`, `1`, `1`, `false`
4. **Reverse a Stack**
   - **Description**: Reverse the elements of a stack.
   - **Input**: Stack: `[1, 2, 3, 4]`
   - **Output**: Stack: `[4, 3, 2, 1]`
5. **Min Stack**
   - **Description**: Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.
   - **Input**: `push(-2)`, `push(0)`, `push(-3)`, `getMin()`, `pop()`, `top()`, `getMin()`
   - **Output**: `null`, `null`, `null`, `-3`, `null`, `0`, `-2`

Medium

1. **Daily Temperatures**
   - **Description**: Given a list of daily temperatures, return a list where each element is the number of days you have to wait until a warmer temperature.
   - **Input**: `[73, 74, 75, 71, 69, 72, 76, 73]`
   - **Output**: `[1, 1, 4, 2, 1, 1, 0, 0]`
2. **Queue Reconstruction by Height**
   - **Description**: Reconstruct the queue based on the given heights and the number of people in front.
   - **Input**: `[[7, 0], [4, 4], [7, 1], [5, 0], [6, 1], [5, 2]]`
   - **Output**: `[[5, 0], [7, 0], [5, 2], [6, 1], [7, 1], [4, 4]]`
3. **Next Greater Element I**
   - **Description**: Find the next greater element for each element in the first array with

respect to the second array.

- **Input**: `nums1 = [4, 1, 2], nums2 = [1, 3, 4, 2]`
- **Output**: `[-1, 3, -1]`

4. **Implement Stack with Min Function**
   - **Description**: Implement a stack with a `min` function that returns the minimum element in the stack.
   - **Input**: `push(5)`, `push(3)`, `min()`, `pop()`, `min()`
   - **Output**: `null`, `null`, `3`, `null`, `5`
5. **Sort a Stack**
   - **Description**: Sort a stack in ascending order using another stack.
   - **Input**: Stack: `[3, 1, 4, 2]`
   - **Output**: Stack: `[1, 2, 3, 4]`

Hard

1. **Longest Valid Parentheses**
   - **Description**: Find the length of the longest valid (well-formed) parentheses substring.
   - **Input**: `"(()())"`
   - **Output**: `6`
2. **Evaluate Reverse Polish Notation**
   - **Description**: Evaluate the value of an arithmetic expression in Reverse Polish Notation.
   - **Input**: `["2", "1", "+", "3", "*"]`
   - **Output**: `9`
3. **Sliding Window Maximum**
   - **Description**: Find the maximum value in each sliding window of size `k`.
   - **Input**: `nums = [1, 3, -1, -3, 5, 3, 6, 7], k = 3`
   - **Output**: `[3, 3, 5, 5, 6, 7]`
4. **Trapping Rain Water**
   - **Description**: Calculate how much water can be trapped after raining given the elevation map.
   - **Input**: `[0,1,0,2,1,0,1,3,2,1,2,1]`
   - **Output**: `6`
5. **Maximal Rectangle**
   - **Description**: Find the area of the largest rectangle that can be formed from a binary matrix.
   - **Input**: `[['1','0','1','0','0'], ['1','0','1','1','1'], ['1','1','1','1','1'], ['1','0','0','1','0']]`
   - **Output**: `6`

These problems cover a range of skills related to stacks and queues, from basic operations to more complex scenarios.

Is this conversation helpful so far?

Upgrade plan
Get GPT-4, DALL·E, and more

Message ChatGPT

ChatGPT can make mistakes. Check important info.