Data Structures & Algorithms Assignment-4

Question: https://leetcode.com/problems/valid-parentheses/description/

Answer: https://leetcode.com/problems/valid-parentheses/submissions/1628647515

Description:

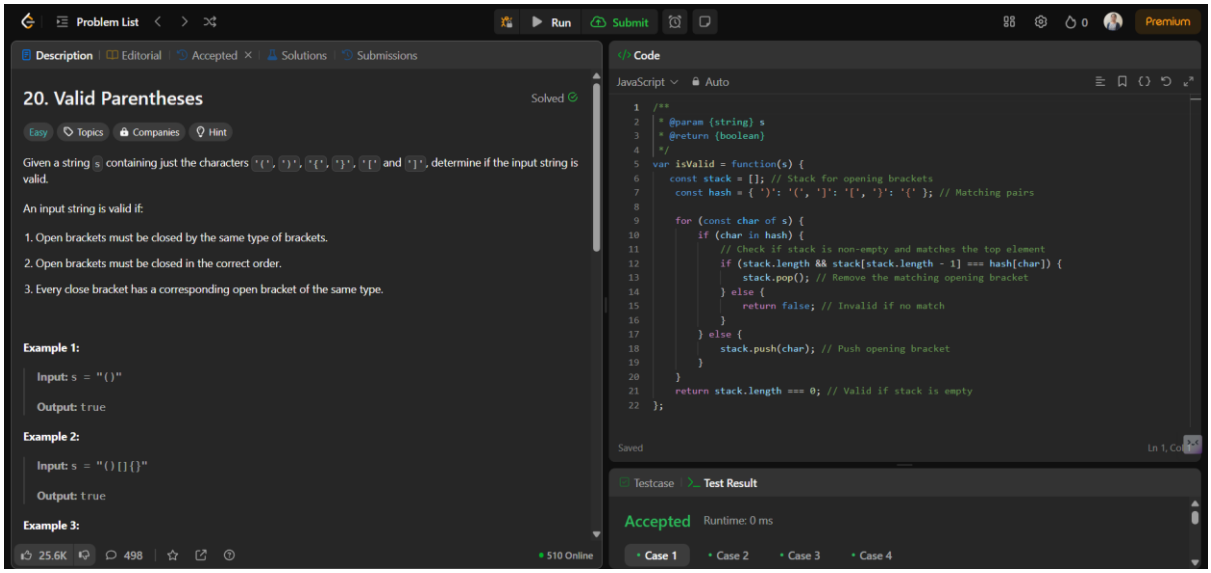
Time Complexity: O(n)

- Where n is the length of the input string s.
- Each character is processed exactly once: either pushed or popped from the stack.

Space Complexity: O(n)

- In the worst case, all characters are opening brackets and get pushed onto the stack.
- So, the space used by the stack grows linearly with the size of the input.

Screenshot:



Question: https://leetcode.com/problems/next-greater-element-i/description/

Answer: https://leetcode.com/problems/next-greater-element-i/submissions/1628654519

Description:

Time Complexity: O(n)

- n = length of nums2
- m = length of nums1
- First loop (over nums2): Each element is pushed and popped from the stack at most once → O(n)
- Second loop (over stack): Fills in -1 for elements with no greater element → at most O(n)
- Final map loop (over nums1): Builds the result array → O(m)

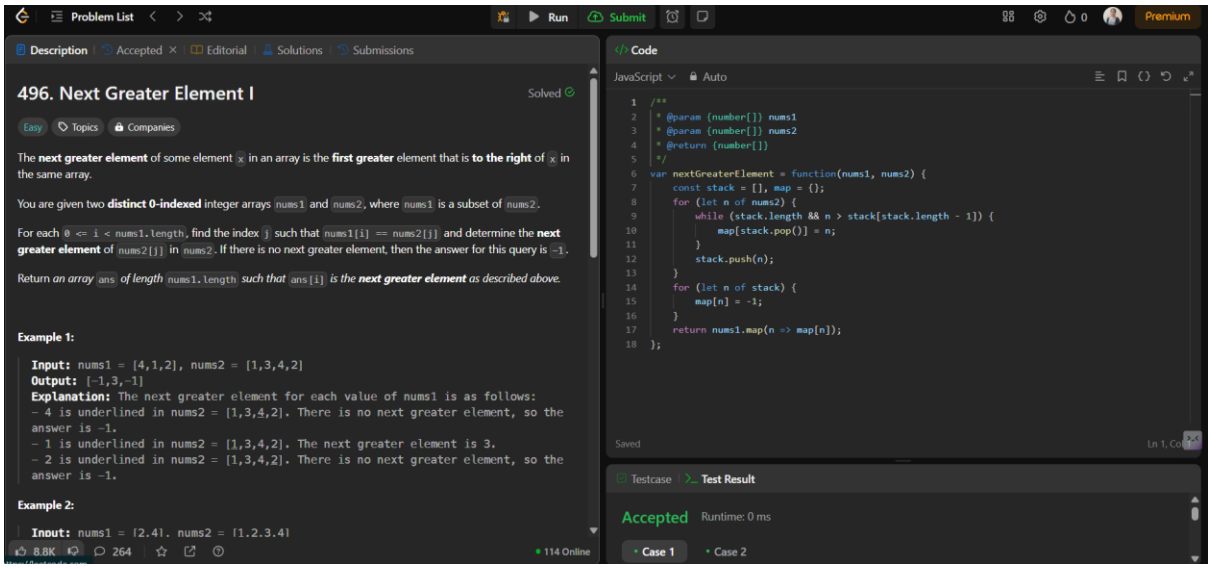
Total: O (n + m)

Space Complexity: O(n)

- map stores a key-value pair for each element in nums2 → O(n)
- stack also holds up to n elements in the worst case → O(n)
- Output array is of size m, but typically not counted against space complexity since it's the return value.

Total auxiliary space: O(n)

Screenshot:



Question: https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/description/

Answer: https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/submissions/1628655814

Description:

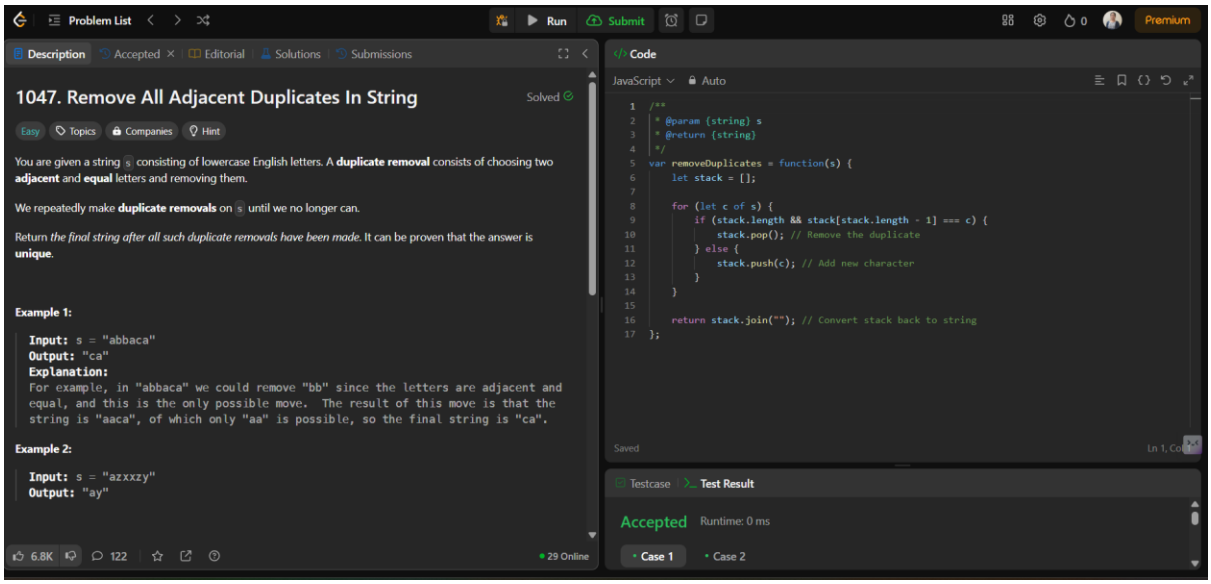
Time Complexity: O(n)

- n is the length of the input string s.
- Each character is pushed and popped at most once, so total operations are linear.

Space Complexity: O(n)

- In the worst case (e.g., no duplicates), all characters go into the stack.
- So, space used by the stack is up to n.

Screenshot:



Question: https://leetcode.com/problems/trapping-rain-water/description/

Answer: https://leetcode.com/problems/trapping-rain-water/submissions/1628657003

Description:

Time Complexity: O(n)

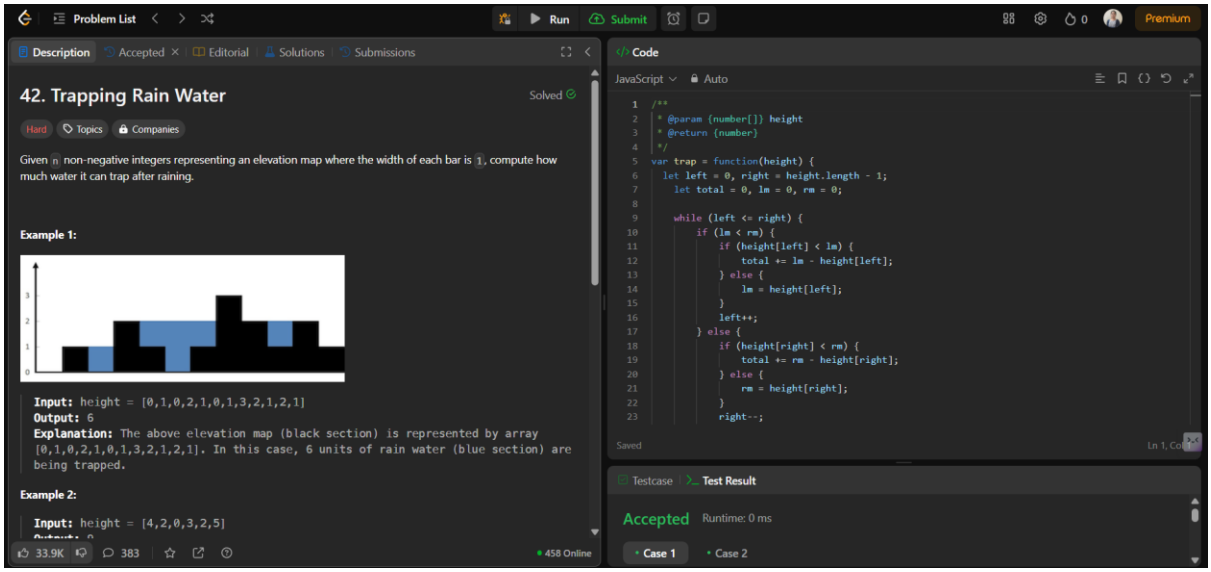
- The array is scanned once using two pointers (left and right).
- Each index is visited at most once.

Total: O(n), where n is the length of the height array

Space Complexity: O(1)

- No extra data structures are used that grow with input size.
- Just a few variables (left, right, lm, rm, total).

Screenshot:



Question: https://leetcode.com/problems/largest-rectangle-in-histogram/description/

Answer: https://leetcode.com/problems/largest-rectangle-in-histogram/submissions/1628657813

Description:

Time Complexity: O(n)

- Each bar is pushed and popped from the stack once.
- All operations inside the loop are constant time.
- So total work is proportional to n.

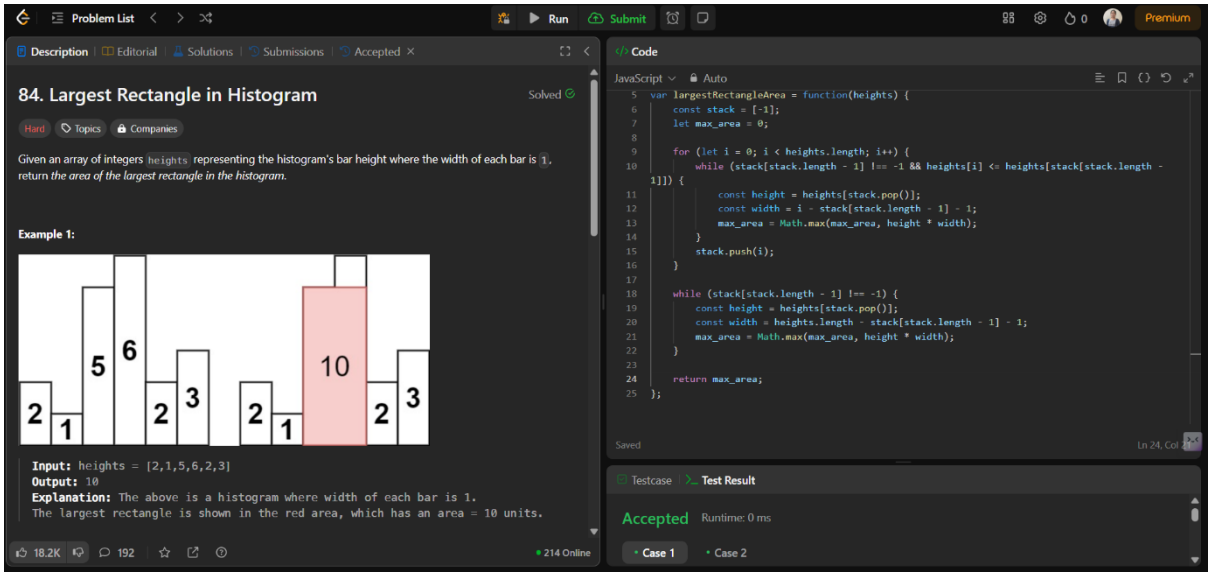
Total: O(n), where n is the number of bars in heights.

Space Complexity: O(n)

- The stack stores up to n indices in the worst case.
- No other significant memory usage.

Total: O(n) auxiliary space.

Screenshot:



Thank You For This Assignment