



[Click on the logo to find **Problem Statement**]

Intuition

The code uses a stack to efficiently determine the number of days until a warmer temperature is encountered for each day in a given list of temperatures. It iterates through the temperatures, maintaining a stack of indices of temperatures for which a warmer day is yet to be found. Whenever a higher temperature is encountered, the algorithm pops indices from the stack, updating the result for each popped index based on the time difference between the current day and the previously stored index. By doing this, it ensures that each temperature's corresponding result is determined as soon as a warmer day is identified. The stack keeps track of indices in a decreasing order of temperatures, optimizing the process for finding the next warmer day. The final result is a list indicating the number of days until a warmer temperature is encountered for each day.

Approach

Approach:

1. Initialization:

- Initialize an answer list with zeros to store the results for each day.
- Initialize an empty stack to keep track of indices of temperatures.

2. Iterative Process:

- Iterate through each temperature in the input list.
- While the stack is not empty and the current temperature is greater than the temperature at the index stored at the top of the stack:
 - Pop the index from the stack.
 - Update the answer for the popped index based on the time difference between the current day and the popped index.
 - Repeat this process until the conditions are no longer met.

3. Updating Stack:

- Push the current index onto the stack.
- Move to the next temperature and repeat the process.

4. Final Result:

- The answer list now contains the number of days until a warmer temperature is encountered for each day.

Complexity

- Time complexity: $O(n)$
- Space complexity: $O(n)$

Code

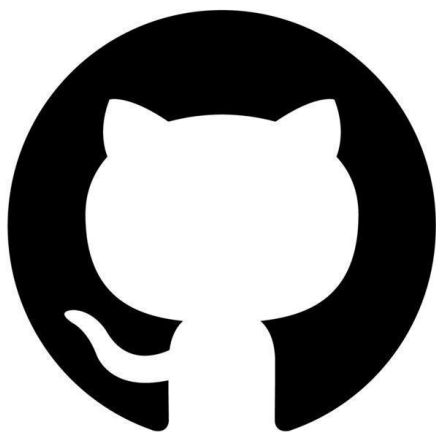
```
class Solution:
    def dailyTemperatures(self, temperatures: List[int]) -> List[int]:
        answer = [0] * len(temperatures)
        stack = []

        for i in range(len(temperatures)):
            while len(stack) > 0 and temperatures[i] >
                temperatures[stack[-1]]:
                prev_index = stack.pop()
                answer[prev_index] = i - prev_index

            stack.append(i)

        return answer
```

If you want to see more solutions to coding problems, you can visit:



GitHub