

Problem List < > ⌂ Submit ⌂ + Premium

Description Accepted × Editorial Solutions Submissions ⌂ ⌂

All Submissions

Accepted 48 / 48 testcases passed

Siddhant0705 submitted at Feb 17, 2026 21:46

Runtime 32 ms Beats 19.38% Memory 107.43 MB Beats 21.93%

Analyze Complexity

Runtime distribution chart showing execution time in milliseconds (5ms to 40ms) versus percentage (0% to 8%).

```
</> Code C++ v Auto
```

```
1 class Solution {
2 public:
3     vector<int> dailyTemperatures(vector<int>& temperatures) {
4
5         int n = temperatures.size();
6         vector<int> result(n, 0);
7         stack<int> st; // store indices
8
9         for(int i = 0; i < n; i++) {
10
11             while(!st.empty() && temperatures[i] > temperatures[st.top()]) {
12                 int prevIndex = st.top();
13                 st.pop();
14                 result[prevIndex] = i - prevIndex;
15             }
16
17             st.push(i);
18         }
19
20     return result;
21 }
22 }
```

Saved Ln 23, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

```
1 class Solution {
2 public:
3     vector<int> dailyTemperatures(vector<int>& temperatures) {
4
5         int n = temperatures.size();
6         vector<int> result(n, 0);
7         stack<int> st; // store indices
8
9         for(int i = 0; i < n; i++) {
10
11             while(!st.empty() && temperatures[i] > temperatures[st.top()]) {
12                 int prevIndex = st.top();
13                 st.pop();
14                 result[prevIndex] = i - prevIndex;
15             }
16
17             st.push(i);
18         }
19
20     return result;
21 }
22 }
```

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 99 / 99 testcases passed
Siddhant0705 submitted at Feb 17, 2026 21:47

Runtime 22 ms Beats 87.70% Memory 92.06 MB Beats 59.11%

Analyze Complexity

Runtime distribution chart showing performance across various execution times.

Code C++

```
public:
stack<pair<int,int>> st;
// {price, span}

StockSpanner() {
}

int next(int price) {
    int span = 1;

    while(!st.empty() && st.top().first <= price) {
        span += st.top().second;
        st.pop();
    }

    st.push({price, span});
    return span;
}
```

Saved Ln 25, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

Case 1

```
1 class StockSpanner {
2 public:
3
4     stack<pair<int,int>> st;
5     // {price, span}
6
7     StockSpanner() {
8 }
```

View more