1. Brute force:
   1. Traverse from begin to end. Pick a temperature day. (index i)
   2. Traverse from next element to end. If have a day warmer (index j), the result is j – i. If not found any warmer day, answer = 0
   3. 2 for loop inside => Complexity is O(n^2)
2. Stack

[](https://leetcode.com/riemeltm/)

✅ [C++] EASY standard sol || INTUITIVE approach with DRY RUN || STACK appraoch

[riemeltm](https://leetcode.com/riemeltm/)



1664

21794

Nov 13, 2021

C

Stack

Hello Everyone, thank you all for reading in advance!

***Note****: Kindly Upvote* 🔼 *if you liked my solution*

**Observation:**

1. For a temperature at index i, we need to find the slightly warmer temperature (at position j ) than the current temperature for 0 <= i < j < n. Which means we need to find the **next greater element** for the current element at index i and if not found then we place 0 in that place.
2. Next Greater Element is a standard problem that is done by using stack and here we just have a slight variation in it. Instead of finding the next greater element, we need to find the distance between current element (temperature) and the next greater element which can be esily done.

**My Appraoch:**

1. We will keep a **stack** and we will start iterating from the end of the temperatures vector (n-1).
2. Now, our main objective is to keep the candidate temperatures on the stack that can be a next greater temperature for current index. (It will become more clear when you'll read more below).
3. Now, for a current temperature at index i, we will see the top of the stack and check whether it is greater than current element or not. **Why we do this?** Because stack is a LIFO and we came from right, so the top of the stack will be the latest element/temperature, that would have been pushed into the stack from the right of current index i so here can be **two cases:**
   * **Case 1:** When this temperature at top of stack is less than the current temperature, then this can never be a next greater element of current temperature or all the other temperatures before current temperature. **Why?** Because for temperatures before current temperature (at position < i), temperature at current index can be their candidate for the next greater element. Hence we will simply pop the top of the stack out until we find an element greater than the current index or until the stack gets empty (This, means no greater element on the right).
   * **Case 2:** When this temperature at top of the stack is greater than the current then we will simply consider this element as our next greater element and set the distance between the current temperature and next greater element/temperature as our answer for current index.
4. After all this is done, then we will have either 0 or the distance between the current temp and next greater temp in the answer vector. Then we will simply push our current element into the stack as now this temp can be a candidate to become the next greater element for the temperatures before this temperature. And then move to the left (iteration goes on like this)

**Note:** I may write nge for next greater element.

**Dry Run:** Suppose our temperatures is like this:  
**Iteration 1:**

nge vector = [0, 0, 0, 0, 0, 0, 0, 0] // initially all set to 0

[73,74,75,71,69,72,76,73]

👆(starting position in the iteration)

Now, our stack is empty that means we don't have anything greater than 73, now we push 73 in our stack and we move left.

**Iteration 2:**

nge vector = [0, 0, 0, 0, 0, 0, 0, 0]

stack = [73] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now, we will check the top of stack until we find the next greater element for 76, but 73 is smaller and after we pop stack then stack gets empty, that means we don't have anything greater than 76 to the right of current index. We will simply push 76 to the stack and move left (since this can be a candidate now).

**Iteration 3:**

nge vector = [0, 0, 0, 0, 0, 0, 0, 0]

stack = [76] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now we check the top of our stack which is 76 and it is greater than 72, so we do not pop it and take it as our next greater element for 72. We then push current temperature to the stack and move left.

**Iteration 4:**

nge vector = [0, 0, 0, 0, 0, 76, 0, 0]

stack = [76, 72] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now, we check the top of the stack which is 72 and it is greater than 69, so we do not pop it and take it as our next greater element for 69. We then push current temperature to the stack and move left.

**Iteration 5:**

nge vector = [0, 0, 0, 0, 72, 76, 0, 0]

stack = [76, 72, 69] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now, we check while the top of the stack is less than or equals to current temp 71 that is we pop it from the stack, so 69 get popped out and we get 72 as our next greater element/temperature. So we take it, and push current temperature to stack and move left.

**Iteration 6:**

nge vector = [0, 0, 0, 72, 72, 76, 0, 0]

stack = [76, 72, 71] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now, we pop top of the stack till we found our next greater element for 75 and hence 71 and 72 gets popped out in the process, so we get 76 as our next greater element for 75 and then we push current temperature to the stack and move left.

**Iteration 7:**

nge vector = [0, 0, 76, 72, 72, 76, 0, 0]

stack = [76, 75] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now the top of stack 75 is greater than 74, so it is our next greater element of our current temperature, so we take it and push current temperature to the stack and move left.

**Iteration 8:**

nge vector = [0, 75, 76, 72, 72, 76, 0, 0]

stack = [76, 75, 74] <- top

[73,74,75,71,69,72,76,73]

👆(current position in iteration)

Now the top of the stack 74 is greater than 73, so we take it as our next greater element and push current temperature to the stack and move left.

**Iteration Complete:** Now the iteration is completeed and we have:

nge vector = [74, 75, 76, 72, 72, 76, 0, 0]

stack = [76, 75, 74, 73] <- top

[73,74,75,71,69,72,76,73]

So our temperatures array and next greater element array looks like this, so the answer array will be just the distance between the current temperature and the next greater element of current temperature

temperatures = [73, 74, 75, 71, 69, 72, 76, 73]

next greater = [74, 75, 76, 72, 72, 76, 0, 0]

ans vector = [1,1,4,2,1,1,0,0]

**Note:** For distance, we will push the index of temperatures in our stack so that we can calculate distance at the same time in the iterations.

Below is the code for my approach mentioned above:

class Solution {

public:

vector<int> dailyTemperatures(vector<int>& temperatures) {

int n = temperatures.size();

vector<int>nge(n, 0); // initially all 0, stores distance between their next greater element and current temperature

stack<int>st{};

// move from right to left

for(int i = n-1; i>=0; --i){

// pop until we find next greater element to the right

// since we came from right stack will have element from right only

// s.top() is the index of elements so we put that index inside temperatures vector to check

while(!st.empty() && temperatures[st.top()] <= temperatures[i])

st.pop();

// if stack not empty, then we have some next greater element,

// so we take distance between next greater and current temperature

// as we are storing indexes in the stack

if(!st.empty())

nge[i] = st.top()-i; // distance between next greater and current

// push the index of current temperature in the stack,

// same as pushing current temperature in stack

st.push(i);

}

return nge;

}

};

**Time Complexity:** ***O(N)*** Since we move from right to left, and the inner while loop stack only visit each temperature of the temperatures vector only once in all the iterations, so it's **O(N)** time complexity (You can observe from dry run)  
**Space Complexity:** ***O(N)*** (stack is used)

Please do UPVOTE 🔼 as it keeps me motivated to make such solution posts