<https://leetcode.com/problems/daily-temperatures/>

**Daily Temperatures**

**Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the ith day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead.**

Example 1:

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

Output: [1,1,4,2,1,1,0,0]

Example 2:

Input: temperatures = [30,40,50,60]

Output: [1,1,1,0]

Example 3:

Input: temperatures = [30,60,90]

Output: [1,1,0]

Constraints:

1 <= temperatures.length <= 105

30 <= temperatures[i] <= 100

**Method 1: (Brute Force)**

For every temp find a next higher temp

Time Complexity: O(n2) *[]*

Space Complexity: O(1) *[]*

**Method 2: (Stack)**

Use stack to store temperatures less than current temperature.

Trick is to store index instead of the numbers.

When a temperature is popped from the stack its corresponding waiting days is calculated and stored in result vector.

Time Complexity: O(n) *[1 push for each temp and atmost 1 pop for each temp]*

Space Complexity: O(n) *[maximum stack size possible]*

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

        int n = temperatures.size(), top;

        vector<int> res(n,0);

        stack<int> s;

        for(int i=0;i<n; i++ ){

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

                res[top] = i - top;

                s.pop();

            }

            s.push(i);

        }

        return res;

    }