

直接用数组方法

```
1  class Solution {
2  public:
3      vector<int> dailyTemperatures(vector<int>& temperatures) {
4
5          // 直接用数组解决
6          int N = temperatures.size();
7          vector<int> result;
8          for (int i = 0; i < N; i++) {
9              int B = 0;
10             //看看后面有几个低温
11             for (int j = i; j < N; j++) {
12                 if (temperatures[j] ≤ temperatures[i]) {
13                     B++;
14                 }
15                 else{
16                     break;
17                 }
18                 if(j==N-1){
19                     B=0;
20                 }
21             }
22             result.push_back(B);
23         }
24         return result;
25     }
26 };````
```

用栈，单调栈

```
32  ````C++
33  class Solution {
34  public:
35      vector<int> dailyTemperatures(vector<int>& temperatures) {
36          // 获取数组的长度
37          int N = temperatures.size();
38          // 结果返回容器，初始化为0
39          vector<int> result(N, 0);
40          // 定义栈，用于存储温度的下标
41          stack<int> temperatures_zhan;
```

```
42         for (int i = 0; i < N; i++) {
43             int t = temperatures[i];
44             // 当栈不为空且当前温度大于栈顶温度时，处理栈顶元素
45             while (!temperatures_zhan.empty() &&
46                 t > temperatures[temperatures_zhan.top()]) {
47                 int prevIndex = temperatures_zhan.top(); // 获取栈
顶元素的下标
48                 temperatures_zhan.pop(); // 弹出栈顶元素
49                 result[prevIndex] = i - prevIndex; // 计算与当前天
的差值
50             }
51             // 每次都记录当前温度的下标
52             temperatures_zhan.push(i);
53         }
54
55         return result; // 返回结果数组
56     }
57 };``
58
59
60
```