

739 每日温度

Label: 数组

请根据每日 气温 列表，重新生成一个列表。对应位置的输出为：要想观测到更高的气温，至少需要等待的天数。如果气温在这之后都不会升高，请在该位置用 0 来代替。

例如，给定一个列表 `temperatures = [73, 74, 75, 71, 69, 72, 76, 73]`，你的输出应该是 `[1, 1, 4, 2, 1, 1, 0, 0]`。

提示：气温 列表长度的范围是 `[1, 30000]`。每个气温的值的均为华氏度，都是在 `[30, 100]` 范围内的整数。

- 迭代

```
class Solution {
    public int[] dailyTemperatures(int[] T) {
        int[] r = new int[T.length];
        for (int i = 0; i < T.length; i++) {
            int day = 1;
            for (int j = i + 1; j < T.length; j++) {
                if (T[i] >= T[j]) {
                    day++;
                } else { // <
                    r[i] = day;
                    break;
                }
            }
        }
        return r;
    }
}
```

- 单调栈

```
class Solution {
    public int[] dailyTemperatures(int[] T) {
        int[] r = new int[T.length];
        Stack<Integer> stack = new Stack<>();

        for (int i = 0; i < T.length; i++) {
            int currT = T[i];
            while (!stack.isEmpty() && T[stack.peek()] < currT) { // 出栈 即可计算出栈元素的结果
                int index = stack.pop();
                r[index] = i - index;
            }
            stack.push(i); // 栈中只用存入 index 即可
        }
        return r;
    }
}
```

- 不用栈 直接实现动态倒序更新，有种动态规划的思想

```
class Solution {
    public int[] dailyTemperatures(int[] T) {
        if (null == T || 0 == T.length) return null;
        int[] result = new int[T.length];

        for (int i = T.length - 2; i >= 0; --i) { // 倒序进行更新
            int j = i + 1;
            while (true) {
                if (T[i] < T[j]) {
                    result[i] = j - i;
                    break;
                } else if (result[j] == 0) {
                    result[i] = 0;
                    break;
                }
                j += result[j];
            }
        }
        return result;
    }
}
```