

执行结果: **通过** 显示详情 > 添加备注

执行用时: **1 ms**, 在所有 Java 提交中击败了 **99.52%** 的用户

内存消耗: **41.6 MB**, 在所有 Java 提交中击败了 **56.09%** 的用户

通过测试用例: **57 / 57**

炫耀一下:

写题解, 分享我的解题思路

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	1 ms	41.6 MB	Java	2022/11/11 10:18	添加备注
通过	49 ms	41.2 MB	Java	2022/07/05 19:55	添加备注

```
1 class Solution {
2     public int[] twoSum(int[] nums, int target) {
3         Map<Integer,Integer> map = new HashMap<>();
4         for(int i = 0;i < nums.length;i++){
5             if(map.containsKey(target-nums[i])){
6                 return new int[]{map.get(target-nums[i]),i};
7             }
8             map.put(nums[i],i);
9         }
10        return new int[0];
11    }
12 }
```

您上次编辑到这里, 代码已从您浏览器本地的临时存储中恢复了 还原默认代码模版

已完成 执行用时: 0 ms

输入 [2,7,11,15]
9

输出 [0,1]

预期结果 [0,1]

差别

执行结果: **通过** [显示详情](#)

▶ 添加备注

执行用时: 2 ms, 在所有 Java 提交中击败了 51.72% 的用户

内存消耗: 39.2 MB, 在所有 Java 提交中击败了 93.76% 的用户

通过测试用例: 92 / 92

炫耀一下;



✎ 写题解，分享我的解题思路

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	2 ms	39.2 MB	Java	2022/11/11 10:47	添加备注

Java

- 智能模式

🕒 模拟面试

i

A

2

9

→

□ □
□ □

```

1 class Solution {
2     public boolean isValid(String s) {
3         int n = s.length();
4         if(n % 2 == 1){
5             return false;
6         }
7         Map<Character,Character> map = new HashMap<>();
8         map.put('(',')');
9         map.put('[,']');
10        map.put('{,'}');
11        Deque<Character> stack = new LinkedList<>();
12        for(int i = 0;i < s.length();i++){
13            char ch = s.charAt(i);
14            if(map.containsKey(ch)){
15                if(stack.isEmpty() || stack.peek() != map.get(ch)){
16                    return false;
17                }
18                stack.pop();
19            }else{
20                stack.push(ch);

```

测试用例

代码执行结果

调试器  Beta

已完成 执行用时: 0 ms

输入

"013"

输出

true

预期结果

true

差别



三 题目列表

随机一题

[< 上一题](#)

20/2841

[下一题 >](#)

控制台 ▲

填入示例 ▼ 如何创建一个测试用例? ▼

► 执行代码 ▼

提交

执行结果: 通过 显示详情 >

添加备注

执行用时: 1 ms, 在所有 Java 提交中击败了 100.00% 的用户

内存消耗: 41.2 MB, 在所有 Java 提交中击败了 21.74% 的用户

通过测试用例: 306 / 306

炫耀一下:



写题解, 分享我的解题思路

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	1 ms	41.2 MB	Java	2022/11/12 14:58	添加备注

```
1 class Solution {
2     public int orangesRotting(int[][] grid) {
3         int M = grid.length;
4         int N = grid[0].length;
5         Queue<int[]> queue = new LinkedList<>();
6
7         int count = 0;
8         for (int r = 0; r < M; r++) {
9             for (int c = 0; c < N; c++) {
10                 if (grid[r][c] == 1) {
11                     count++;
12                 } else if (grid[r][c] == 2) {
13                     queue.add(new int[]{r, c});
14                 }
15             }
16         }
17
18         int round = 0;
19         while (count > 0 && !queue.isEmpty()) {
20             round++;
21             int n = queue.size();
22             for (int i = 0; i < n; i++) {
23                 int[] cell = queue.poll();
24                 int r = cell[0], c = cell[1];
25                 if (grid[r][c] == 2) continue;
26                 if (r > 0 && grid[r-1][c] == 1) {
27                     queue.add(new int[]{r-1, c});
28                     grid[r-1][c] = 2;
29                 }
30                 if (r < M-1 && grid[r+1][c] == 1) {
31                     queue.add(new int[]{r+1, c});
32                     grid[r+1][c] = 2;
33                 }
34                 if (c > 0 && grid[r][c-1] == 1) {
35                     queue.add(new int[]{r, c-1});
36                     grid[r][c-1] = 2;
37                 }
38                 if (c < N-1 && grid[r][c+1] == 1) {
39                     queue.add(new int[]{r, c+1});
40                     grid[r][c+1] = 2;
41                 }
42             }
43             count = 0;
44             for (int r = 0; r < M; r++) {
45                 for (int c = 0; c < N; c++) {
46                     if (grid[r][c] == 1) count++;
47                 }
48             }
49         }
50         return round;
51     }
52 }
```

测试用例 代码执行结果 调试器 Beta

已完成 执行用时: 0 ms

输入 [[2,1,1],[1,1,0],[0,1,1]]

输出 4

预期结果 4



题目列表

随机一题

< 上一题 994/2841

下一题 >

控制台 填入示例 如何创建一个测试用例?

执行代码

提交

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	4 ms	41.3 MB	Java	2022/11/12 15:39	添加备注
解答错误	N/A	N/A	Java	2022/11/12 15:36	添加备注
解答错误	N/A	N/A	Java	2022/11/12 15:21	添加备注

```
1 class Solution {
2     public int lengthOfLongestSubstring(String s) {
3         if (s.length() == 0) return 0;
4         Map<Character, Integer> map = new HashMap<>();
5         int max = 0;
6         int l = 0;
7         for (int i = 0; i < s.length(); i++) {
8             if (map.containsKey(s.charAt(i))) {
9                 l = Math.max(l, map.get(s.charAt(i)) + 1);
10            }
11            map.put(s.charAt(i), i);
12            max = Math.max(max, i - l + 1);
13        }
14        return max;
15    }
16 }
```

您上次编辑到这里，代码已从您浏览器本地的临时存储中恢复了 还原默认代码模版

执行结果: **通过** [显示详情](#) [添加备注](#)

执行用时: **51 ms**, 在所有 Java 提交中击败了 **66.72%** 的用户

内存消耗: **53.3 MB**, 在所有 Java 提交中击败了 **51.17%** 的用户

通过测试用例: **160 / 160**

炫耀一下:

[写题解, 分享我的解题思路](#)

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	51 ms	53.3 MB	Java	2022/11/12 16:32	添加备注
解答错误	N/A	N/A	Java	2022/11/12 16:29	添加备注

```
1 class Solution {
2     public long totalCost(int[] costs, int k, int candidates) {
3         PriorityQueue<Integer> q1 = new PriorityQueue<Integer>();
4         PriorityQueue<Integer> q2 = new PriorityQueue<Integer>();
5         int l = candidates, n = costs.length, r = Math.max(l - 1, n - candidates - 1);
6         long ans = 0;
7         for (int i = 0; i < candidates; i++) q1.add(costs[i]);
8         for (int i = n - 1; i > r; i--) q2.add(costs[i]);
9         while (k-- > 0) {
10             if (!q1.isEmpty() && !q2.isEmpty()) {
11                 if (q1.peek() <= q2.peek()) {
12                     ans += q1.poll();
13                     if (l <= r) q1.add(costs[l++]);
14                 } else {
15                     ans += q2.poll();
16                     if (l <= r) q2.add(costs[r--]);
17                 }
18             } else if (q2.isEmpty()) {
19                 ans += q1.poll();
20                 if (l <= r) q1.add(costs[l++]);
21             } else if (q1.isEmpty()) {
22                 ans += q2.poll();
23                 if (l <= r) q2.add(costs[r--]);
24             }
25         }
26         return ans;
27     }
28 }
29 }
```