

```

1  /*
2  给定一个字符串 S 和一个字符 C。返回一个代表字符串 S 中每个字符到字符串 S 中的字符 C 的最
   短距离的数组。
3
4  示例 1:
5
6  输入: S = "loveleetcode", C = 'e'
7  输出: [3, 2, 1, 0, 1, 0, 0, 1, 2, 2, 1, 0]
8  说明:
9
10 字符串 S 的长度范围为 [1, 10000]。
11  C 是一个单字符,且保证是字符串 S 里的字符。
12  S 和 C 中的所有字母均为小写字母。
13
14 来源: 力扣 (LeetCode)
15 链接: https://leetcode-cn.com/problems/shortest-distance-to-a-character
16 著作权归领扣网络所有。商业转载请联系官方授权,非商业转载请注明出处。
17 */

```

分析:

- 暴力法:
 - 首选遍历一次S,找到所有C的索引
 - 再遍历一次S,然后将获取距离得最小值

方法一:C++_暴力法

```

1  class solution
2  {
3
4      public:
5          vector<int> shortestToChar(string S, char C)
6          {
7              vector<int> ret_val      ;
8              vector<int> temp_ind     ;
9              int i = 0 ;
10             int j = 0 ;
11             int min_ind = 0 ;
12
13             /*找到S中所有C的索引*/
14             for(i=0;i<S.size();i++)
15             {
16                 if(C==S.at(i))
17                 {
18                     temp_ind.push_back(i);
19                 }
20             }
21
22             /*遍历S中的每一个元素并计算*/
23             for(i=0;i<S.size();i++)
24             {
25                 min_ind = 10000;

```

```
26         if(S.at(i)==C)
27         {
28             min_ind = 0;
29         }
30         else
31         {
32             for(j=0;j<temp_ind.size();j++)
33             {
34                 if(abs( temp_ind[j] - i ) < min_ind)
35                 {
36                     min_ind = abs( temp_ind[j] - i );
37                 }
38             }
39         }
40         ret_val.push_back(min_ind);
41     }
42     return ret_val;
43 }
44 };
45 /*
46 执行结果:
47 通过
48 显示详情
49 执行用时 :20 ms, 在所有 C++ 提交中击败了47.42%的用户
50 内存消耗 :8.9 MB, 在所有 C++ 提交中击败了74.80%的用户
51 */
```