821 Shortest Distance to a Character

2018年5月5日 20:37

Given a string S and a character C, return an array of integers representing the shortest distance from the character C in the string.

Example 1:

Input: S = "loveleetcode", C = 'e'

Output: [3, 2, 1, 0, 1, 0, 0, 1, 2, 2, 1, 0]

Note:

1. S string length is in [1, 10000].
2. C is a single character, and guaranteed to be in string S.
3. All letters in S and C are lowercase.

来自 <https://leetcode.com/problems/shortest-distance-to-a-character/description/>

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。

示例 1:

输入: S = "loveleetcode", C = 'e'

输出: [3, 2, 1, 0, 1, 0, 0, 1, 2, 2, 1, 0]

说明:

1. 字符串 S 的长度范围为 [1, 10000]。
2. C 是一个单字符，且保证是字符串 S 里的字符。
3. S 和 C 中的所有字母均为小写字母。

Solution for Python3:

```python
class Solution1:
    def shortestToChar(self, S, C):
        """
        :type S: str
        :type C: str
        :rtype: List[int]
        """
        prev = float('-inf')
        ans = []
        for i, x in enumerate(S):
            if x == C:
                prev = i
            ans.append(i - prev)
        prev = float('inf')
        for i in range(len(S) - 1, -1, -1):
            if S[i] == C:
                prev = i
```

```
            ans[i] = min(ans[i], prev - i)
        return ans

class Solution2:
    def shortestToChar(self, S, C):
        """
        :type S: str
        :type C: str
        :rtype: List[int]
        """
        n = len(S)
        ans = [n] * n
        prev = n
        for i in list(range(n)) + list(range(n))[::-1]:
            if S[i] == C:
                prev = i
            ans[i] = min(ans[i], abs(prev - i))
        return ans
```

Solution for C++:

```
// Explanation
// Initial result array.
// Loop twice on the string S.
// First loop find shortest distant to character on left.
// Second loop find shortest distant to character on right.
class Solution {
public:
    vector<int> shortestToChar(string S, char C) {
        vector<int> res;
        int prev = -10000;
        for (int i = 0; i < S.length(); i++) {
            if (S[i] == C)
                prev = i;
            res.push_back(i - prev);
        }
        prev = 100000;
        for (int i = S.length() - 1; i >= 0; i--) {
            if (S[i] == C)
                prev = i;
            res[i] = min(res[i], prev - i);
        }
        return res;
```

```
23          }
24      };
```