←(/) Explore(/explore/) Problems(/problemset/all/) Interview ^New Contest ^(/contest/) Discuss(/discuss/) ▯ (https://leetcode.com/subscribe/?...Storeref=nb_npl&source=nav-tag) ☆ Premium (/subscribe?ref=nb_npl) 🔔 ◊ 0profit-i

schedu

6250. Minimum Penalty for a Shop

| My Submissions (/contest/biweekly-contest-92/problems/minimum-penalty-for-a-shop/submissions/) | Back to Contest (/contest/biweekly-contest-92/) |

You are given the customer visit log of a shop represented by a **0-indexed** string `customers` consisting only of characters `'N'` and `'Y'` :

- if the ith character is `'Y'` , it means that customers come at the ith hour
- whereas `'N'` indicates that no customers come at the ith hour.

If the shop closes at the jth hour (`0 <= j <= n`), the **penalty** is calculated as follows:

- For every hour when the shop is open and no customers come, the penalty increases by `1` .
- For every hour when the shop is closed and customers come, the penalty increases by `1` .

Return the **earliest** hour at which the shop must be closed to incur a **minimum** penalty.

Note that if a shop closes at the jth hour, it means the shop is closed at the hour `j` .

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: customers = "YYNY"
Output: 2
Explanation:
- Closing the shop at the 0th hour incurs in 1+1+0+1 = 3 penalty.
- Closing the shop at the 1st hour incurs in 0+1+0+1 = 2 penalty.
- Closing the shop at the 2nd hour incurs in 0+0+0+1 = 1 penalty.
- Closing the shop at the 3rd hour incurs in 0+0+1+1 = 2 penalty.
- Closing the shop at the 4th hour incurs in 0+0+1+0 = 1 penalty.
Closing the shop at 2nd or 4th hour gives a minimum penalty. Since 2 is earlier, the optimal closing time is 2.
```

Example 2:

```
Input: customers = "NNNNN"
Output: 0
Explanation: It is best to close the shop at the 0th hour as no customers arrive.
```

Example 3:

```
Input: customers = "YYYY"
Output: 4
Explanation: It is best to close the shop at the 4th hour as customers arrive at each hour.
```

Constraints:

- `1 <= customers.length <= 10^5`
- `customers` consists only of characters `'Y'` and `'N'` .

JavaScript ▾

```javascript
 1 ▾ function Bisect() {
 2       return { insort_right, insort_left, bisect_left, bisect_right }
 3 ▾     function insort_right(a, x, lo = 0, hi = null) {
 4           lo = bisect_right(a, x, lo, hi);
 5           a.splice(lo, 0, x);
 6       }
 7 ▾     function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
 8           if (lo < 0) throw new Error('lo must be non-negative');
 9           if (hi == null) hi = a.length;
10 ▾         while (lo < hi) {
11               let mid = parseInt((lo + hi) / 2);
12               a[mid] > x ? hi = mid : lo = mid + 1;
13           }
14           return lo;
15       }
16 ▾     function insort_left(a, x, lo = 0, hi = null) {
```

```
17          lo = bisect_left(a, x, lo, hi);
18          a.splice(lo, 0, x);
19      }
20 ▾   function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
21          if (lo < 0) throw new Error('lo must be non-negative');
22          if (hi == null) hi = a.length;
23 ▾       while (lo < hi) {
24              let mid = parseInt((lo + hi) / 2);
25              a[mid] < x ? lo = mid + 1 : hi = mid;
26          }
27          return lo;
28      }
29 }
30
31 ▾ const bestClosingTime = (s) => {
32      let n = s.length, a = [], b = [], d = [], bi = new Bisect();
33      for (let i = 0; i < n; i++) s[i] == 'Y' ? a.push(i) : b.push(i);
34 ▾   for (let j = 0; j <= n; j++) {
35          let leftN = bi.bisect_right(b, j - 1);
36          let rightY = a.length - bi.bisect_left(a, j);
37          let cnt = leftN + rightY;
38          d.push([cnt, j]);
39      }
40 ▾   d.sort((x, y) => {
41          if (x[0] != y[0]) return x[0] - y[0];
42          return x[1] - y[1];
43      })
44      return d[0][1]
45 };
```

☐ **Custom Testcase**   (Use Example Testcases)

▶ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/850157607/) ❓   (More Details ❯ (/submissions/detail/850157607/))

Share your acceptance!