**2483. Minimum Penalty for a Shop :-**

Medium Accepted: 49K Submissions: 73.4K Acceptance Rate: 66.7%

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.

**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 <= 105
* customers consists only of characters 'Y' and 'N'.

**Code :-**

class Solution {

public:

    int bestClosingTime(string customer) {

        int n=customer.size();

        vector<pair<int, int>> pre(n);

        customer[0] == 'Y' ? pre[0].first=1 : pre[0].first=0;

        customer[0] == 'N' ? pre[0].second=1 : pre[0].second=0;

        int sumy=0, sumn=0;

        for(int i=1; i<n; ++i){

            // for "Y"

            pre[i].first = pre[i-1].first + (customer[i]=='Y' ? 1 : 0);

            // for "N"

            pre[i].second = pre[i-1].second + (customer[i]=='N' ? 1 : 0);

        }

        int penalty=INT\_MAX, ans=-1, cur\_penalty;

        for(int i=0; i<n; ++i){

            int leftno = pre[i].second;

            int rightyes = pre[n-1].first - pre[i].first;

            if(customer[i] == 'N')

                cur\_penalty = leftno - 1 + rightyes;

            else

                cur\_penalty = leftno + rightyes + 1;

            if(cur\_penalty < penalty){

                ans = i;

                penalty = cur\_penalty;

            }

        }

        // explicitly for last calculation

        if(pre[n-1].second < penalty)

            return n;

        return ans;

    }

};

**T.C :- O(N)**

**S.C :- O(N)**