Assalammualaikum

Mohammad Hanif Furgan Aufa Putra

5025221161

PERBAIKAN TUGAS 4 PBO C

T4_MEDIUM_01- Thể chất nhưng mà nó lạ lắm? no tags

In the beautiful KMA kingdom, while everyone is focusing on coding to prepare for the upcoming Hackathon, Son- one of the many unfortunate citizens here- has to face a life-or-death test: the Physical Education re-exam.

Perhaps everyone knows that the final exam for Physical Education I involves a sprint. However, for those retaking the exam, the test is no longer as simple as that!

At the starting point, Ms. Hoa sets up a signal light consisting of three colors: green, red, and yellow (x, d, v). Participants are only allowed to start when the signal light turns green!

Specifically:

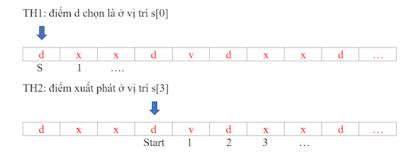
The light will repeat the signal every N seconds, and at the i-th second, the light will change to the si color.

The color of the signal light is represented by a character string.

For example, if the signal string is s = "dxxdv", then the signal light will operate according to the principle: "red-green-green-red-yellow-red-green-green-red-yellow-red-green-green-red-yellow..." and continue to repeat like that.

After understanding the pattern of the signal light color change, Son needs to calculate the minimum time he must wait to start in the worst-case scenario when the signal light starts at its worst color!

With the input string s = "dxxdv" and the current signal light being red (d), there are two cases: the light will turn green after 1 second or after 3 seconds. The answer is 3 seconds - the minimum waiting time in the worst-case scenario.



Input:
The first line is the number of test cases t $(1 \le t \le 10^4)$.
The first line of each test case is an integer n and a character c ($1 \le n \le 2.10^5$, c is one of the valid signal light colors d, v, or x), corresponding to the length of the signal string s and the current signal light.
The second line is a signal string s with a length of n, consisting of characters d, v , and x .
Test cases are always guaranteed to have the following:
There is always the character 'x' and the current signal c in the string s.
Subtask 1: 25% of the test cases have $t \le 102$ and the length n of the signal string $s \le 10^5$.
Subtask 2: The remaining 75% of the test cases have data as stated in the problem statement.
Output:
There are t lines, each line contains an integer which is the minimum waiting time.
Example: Input:
6 1 x x 3 d ddx 5 v vddxv 7 d vxdxvdx 5 d vxxdv 9 v dddxvvxxv
Output:
0 2 4 1 3 4

Masalah ini melibatkan sebuah string karakter yang merepresentasikan lampu isyarat dengan tiga warna: hijau ('x'), merah ('d'), dan kuning ('v'). Lampu isyarat tersebut berulang polanya setiap N detik, dan peserta hanya diperbolehkan memulai ketika lampu isyarat berubah menjadi warna hijau. Tujuan dari masalah ini adalah untuk menghitung waktu minimum yang harus ditunggu peserta sebelum memulai dalam skenario terburuk, yakni ketika lampu isyarat berawal dari warna terburuk ('d' atau 'v').

Untuk menyelesaikan masalah ini, kita perlu memahami pola perubahan warna lampu isyarat, mengidentifikasi posisi 'go' (warna dari mana peserta dapat mulai) dan 'goal' (warna yang harus dicapai) dalam string isyarat, dan selanjutnya menghitung waktu minimum yang harus ditunggu dalam skenario terburuk berdasarkan posisi-posisi tersebut.

Kode yang diberikan dapat memecahkan masalah ini dengan melakukan langkah-langkah berikut. Pertama, kode membaca jumlah kasus uji dan memproses masing-masingnya. Selanjutnya, kode mengidentifikasi warna 'go' dan 'goal', serta mencatat posisi-posisinya dalam string isyarat. Kemudian, kode menghitung waktu minimum yang harus ditunggu berdasarkan posisi-posisi yang telah tercatat.

Kode ini menggunakan vektor 'memgo' dan 'memgoal' untuk menyimpan posisi 'go' dan 'goal' secara berurutan. Dalam setiap iterasi melalui 'memgo', kode mencari posisi 'goal' terdekat yang muncul setelahnya dalam string isyarat. Waktu tunggu dihitung sebagai selisih antara posisi 'goal' terdekat dan posisi 'go' saat itu. Kode selalu melacak waktu tunggu maksimum di antara semua posisi 'go', yang pada akhirnya menggambarkan waktu minimum yang harus ditunggu dalam skenario terburuk.

Dengan demikian, kode ini menghitung waktu minimum yang dibutuhkan untuk memulai dalam situasi terburuk ketika lampu isyarat dimulai dari warna terburuk, dan ini diulang untuk setiap kasus uji.

Dan inilah kode yang saya buat

```
#include <stdio.h>
#include <vector>

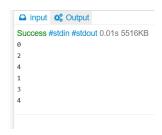
using namespace std;

#define gc getchar_unlocked

template <typename T>
void Getnum(T &val) {
    char ch;
    bool bo = 0;
    val = 0;
    for (ch = gc(); ch < '0' || '9' < ch; ch = gc())
        if (ch == '-')
        bo = 1;
    for (; '0' <= ch && ch <= '9'; val = (val << 3) + (val << 1) + ch - 48, ch = gc());
    if (bo)
    val = -val;
}</pre>
```

```
21 int main() {
        int t;
        Getnum(t); // Read the number of test cases.
        while (t--) {
            char go, goal = 'x';
vector<int> memgo, memgoal;
            int size;
            Getnum(size); // Read the size of the signal string.
            go = getchar_unlocked();
            char s[size + 1];
            scanf("%s", s); // Read the signal string.
            if (go == goal) {
    printf("0\n");
                 continue;
                 if (s[i] == go)
                     memgo.push_back(i); // Record the positions of the 'go' signal.
                 if (s[i] == goal)
                     memgoal.push_back(i); // Record the positions of the 'goal' signal.
            int overall = 0;
            for (int i = 0; i < memgo.size(); i++) {</pre>
                 int nearestgoal = -1;
for (int j = 0; j < memgoal.size(); j++) {</pre>
                     if (memgoal[j] > memgo[i]) {
    nearestgoal = memgoal[j];
                          break;
                 if (nearestgoal == -1)
                     nearestgoal = memgoal[0] + size;
                 int dist = nearestgoal - memgo[i];
                 overall = max(overall, dist); // Calculate the minimum waiting time.
            printf("%d\n", overall); // Print the result.
            memgo.clear();
            memgoal.clear();
```

Dengan menjalan program tersebut diperoleh hasil seperti dibawah ini



Dan dibawah bukti Verdict Accept

