1324C - Frog Jumps

Tutorial

1324C - Frog Jumps

The only observation we need is that we don't need to jump left at all. This only decreases our position so we have less freedom after the jump to the left. Then, to minimize d, we only need to jump between the closest 'R' cells. So, if we build the array $b = [0, r_1, r_2, \ldots, r_k, n+1]$, where r_i is the position of the i-th 'R' cell from left to right (1-indexed), then the answer is $\max\limits_{i=0}^{k} b_{i+1} - b_i$.

Time complexity: $O(n)$.

Solution

```
#include <bits/stdc++.h>

using namespace std;

int main() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
//      freopen("output.txt", "w", stdout);
#endif

        int t;
        cin >> t;
        while (t--) {
                string s;
                cin >> s;
                vector<int> pos;
                pos.push_back(0);
                for (int i = 0; i < int(s.size()); ++i) {
                        if (s[i] == 'R') pos.push_back(i + 1);
                }
                pos.push_back(s.size() + 1);
                int ans = 0;
                for (int i = 0; i < int(pos.size()) - 1; ++i) {
                        ans = max(ans, pos[i + 1] - pos[i]);
                }
                cout << ans << endl;
        }

        return 0;
}
```