PRACTICE CERTIFICATION NEW COMPETE JOBS LEADERBOARD Search hackerrank682

Practice > Algorithms > Strings > Strong Password > Editorial

Strong Password ☆

22.53 more points to get your gold badge!

Rank: **148558** | Points: **827.47/850**

Problem Submissions Leaderboard Discussions **Editorial**

Editorial by anveshi

Problem

Given a password, find the minimum number of characters to add to make it strong.

Solution

The answer is always $\max(6-n, 4-d)$ where n is string length and d is the number of different type of characters that are already present in the input password.

Set by anveshi

Problem Setter's code:

Solution 1

```
#include "bits/stdc++.h"
using namespace std;

const int N = 200005;

int get(char c) {
   if(c >= '0' && c <= '9') return 0;
   if(c >= 'a' && c <= 'z') return 1;
   if(c >= 'A' && c <= 'Z') return 2;
   return 3;
}

int main() {
   int n;
   string s;

   cin >> n;
   cin >> s;

   int ans = 0;

   for(int i = 0; i < n; i++) {
      ans |= (1 << get(s[i]));
   }

   cout << max(6 - n, 4 - __builtin_popcount(ans));
   return 0;
}
```

Solution 2

```
#include "bits/stdc++.h"
using namespace std;

const int N = 200005;

const string numbers = "0123456789";
const string lower_case = "abcdefghijklmnopqrstuvwxyz";
const string upper_case = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

STATISTICS

Difficulty:	Easy
Time Complexity:	$O(n)$
Required Knowledge:	Strings, Loops
Publish Date:	Sep 17 2017

Originally featured in HourRank 24

NEED HELP?

View discussions

View top submissions