SOLVE LATER

A Needle in the Haystack

Attempted by: **2542** / Accuracy: **37%** / Maximum Score: **30** / 22 Votes

Tag(s): Ad-Hoc, Algorithms, Medium

I would like to divide the approach into two parts.

1. Understanding the permutation part : Two really important things to note about permutations are that all permutations of one string have same frequency of any alphabet and that permutations of any string covers all the possible patterns using the given frequency of alphabets. So, instead of actually finding all permutations, what we can do is count the frequency of alphabets in the needle string. This would help us avoid generating all permutations and we can still check for occurrence of any permutation by checking if the frequency of needle is equal to frequency of any part of the haystack string.

2. Matching frequency of needle with haystack : Calculate initially the frequency of alphabets of haystack string from index 0 to strlen(needle) - 1. This gives you a basic data of the frequency. From this, loop from index i = strlen(needle) to strlen(haystack) - 1. At the beginning of the loop, check if the frequency of needle and haystack matches. If it does, break out of the loop and print "YES". Otherwise, decrease the frequency of alphabet at index (i-strlen(needle)) of haystack string by 1 and increase the frequency of alphabet at index i by 1. This way, you can avoid recalculations and use the results found earlier to form a faster solution.

In case you dont understand some part of the editorial, try going through my AC code below. If you still don't understand some part, feel free to comment on this and I would like to explain my approach in even more detail.

My Code in C++ :

```
#include <iostream>
#include <cstring>
#include <vector>
using namespace std;

int main()
{
    int t;
    cin>>t;
    char c;
    cin.get(c);
    while(t--)
    {
        char p[10002],hay[101000];
        cin>>p>>hay;

        int lp = strlen(p);
        int lh = strlen(hay);

        int p_abc[26] = {0}; // stores the frequency of each
character in needle.
        int f = 0;
```

BEST SUBMISSIONS

LANGUAGE: C++

TIME (sec) 1.0048 MEMORY (KiB) 64

by Vivek Kumar Yadav

VIEW BEST SUBMISSION VIEW ALL SUBMISSION

CONTRIBUTOR

AUTHOR
Chandan Singh

THIS PROBLEM WAS ASKED IN

CHALLENGE NAME
IndiaHacks 2014

SOCIAL SHARE

```
        for(int i=0;i<lp;i++)
            p_abc[p[i]-'a']++;

        int checker_abc[26] = {0}; // stores frequency of each
character in haystack

        for(int i=0;i<lp;i++)
        {
            checker_abc[hay[i]-'a']++;
        }

        for(int i=lp;i<lh;i++)
        {
            //first check if found;
            int found = 1;
            for(int k=0;k<26;k++)
            {
                if(p_abc[k] != checker_abc[k])
                {
                    found = 0;break;
                }
            }
            if(found)
            {
                f=1;break;
            }

            checker_abc[hay[i-lp]-'a']--;
            checker_abc[hay[i]-'a']++;
        }
        int found = 1;
        for(int k=0;k<26 && !f;k++)
            {
                if(p_abc[k] != checker_abc[k])
                {
                    found = 0;break;
                }
            }
            if(found) f=1;

        if(!f)
        cout<<"NO"<<endl;
        else
        cout<<"YES"<<endl;
    }
    return 0;
}
```

Submitted by Himanshu Jaju

IS THIS EDITORIAL HELPFUL?

Yes, it's helpful

No, it's not helpful

24 developer(s) found this editorial helpful.