Weighted Uniform Strings

by nabila_ahmed

Problem Submissions Leaderboard Discussions Editorial

A weighted string is a string of lowercase English letters where each letter has a *weight* in the inclusive range from 1 to 26, defined below:

a	1
b	2
c	3
d	4
e	5
f	6
g	7
h	8
i	9
j	10

k	11
l	12
m	13
n	14
o	15
p	16
q	17
r	18

s	19
t	20
u	21
v	22
w	23
x	24
y	25
z	26

We define the following terms:

- The *weight of a string* is the sum of the weights of all the string's characters. For example:

apple	1 + 16 + 16 + 12 + 5 = 50
hack	8 + 1 + 3 + 11 = 23
watch	23 + 1 + 20 + 3 + 8 = 53
ccccc	3 + 3 + 3 + 3 + 3 = 15
aaa	1 + 1 + 1 = 3
zzzz	26 + 26 + 26 + 26 = 104

- A *uniform string* is a string consisting of a single character repeated zero or more times. For example, `ccc` and `a` are uniform strings, but `bcb` and `cd` are not (i.e., they consist of more than one distinct character).

Given a string, s, let U be the set of weights for all possible uniform substrings (contiguous) of string s. You have to answer n queries, where each query i consists of a single integer, x_i. For each query, print `Yes` on a new line if $x_i \in U$; otherwise, print `No` instead.

Note: The $\in$ symbol denotes that x_i is an element of set U.

Input Format

The first line contains a string denoting s (the original string).
The second line contains an integer denoting n (the number of queries).
Each line i of the n subsequent lines contains an integer denoting x_i (the weight of a uniform subtring of s that may or may not exist).

Constraints

- $1 \leq |s|, n \leq 10^5$
- $1 \leq x_i \leq 10^7$

- s will only contain lowercase English letters.

Output Format

Pring n lines. For each query, print `Yes` on a new line if $x_i \in U$; otherwise, print `No` instead.

Sample Input 0

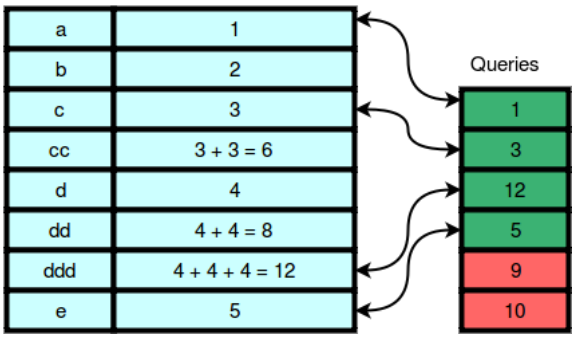
```
abccddde
6
1
3
12
5
9
10
```

Sample Output 0

```
Yes
Yes
Yes
Yes
No
No
```

Explanation 0

The weights of every possible *uniform substring* in the string `abccddde` are shown below:



We print `Yes` on the first four lines because the first four queries match weights of uniform substrings of s. We print `No` for the last two queries because there are no uniform substrings in s that have those weights.

Note that while `de` is a substring of s that would have a weight of 9, it is *not a uniform substring*.

Note that we are only dealing with contiguous substrings. So `ccc` is not a substring of the string `ccxxc`.

Submissions: 1204
Max Score: 20
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  class Solution {
6
```

```
7
8       static Dictionary<char, long> maxUniformString(string s)
9           {
10
11              Dictionary<char, long> diccio = new Dictionary<char, long>();
12
13              for (char ch = 'a'; ch <= 'z'; ch++)
14              {
15                  diccio[ch] = 0;
16              }
17
18              for (int i = 0; i < s.Length; )
19              {
20                  long cont = 0;
21                  char actual = s[i];
22                  while (i < s.Length && s[i] == actual)
23                  {
24                      i++;
25                      cont++;
26                  }
27                  diccio[actual] = Math.Max(cont, diccio[actual]);
28              }
29
30              return diccio;
31          }
32
33          static void Main(string[] args)
34          {
35              string s =  Console.ReadLine();
36              long n = long.Parse(Console.ReadLine());
37
38              Dictionary<char, long> diccio = maxUniformString(s);
39
40              while (n-- > 0)
41              {
42                  long x =  long.Parse(Console.ReadLine());
43
44                  string ans = "No";
45                  for (char ch = 'a'; ch <= 'z'; ch++)
46                  {
47                      if (x % (ch - 'a' + 1) == 0 && x <= diccio[ch]* (ch-'a'+1))
48                      {
49                          ans = "Yes";
50                          break;
51                      }
52                  }
53                  Console.WriteLine(ans);
54              }
55
56              Console.ReadLine();
57          }
58
59
60
61  }
62
```

Line: 57 Col: 10

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5

- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15
- ✔ Test Case #16
- ✔ Test Case #17
- ✔ Test Case #18
- ✔ Test Case #19
- ✔ Test Case #20
- ✔ Test Case #21
- ✔ Test Case #22
- ✔ Test Case #23
- ✔ Test Case #24
- ✔ Test Case #25
- ✔ Test Case #26
- ✔ Test Case #27
- ✔ Test Case #28
- ✔ Test Case #29
- ✔ Test Case #30

Next Challenge