

{% note info %} **摘要** Title: 831. KMP字符串 Tag: KMP、字符串哈希 Memory Limit: 64 MB Time Limit: 1000 ms  
{% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

## 831. KMP字符串

### • 题意

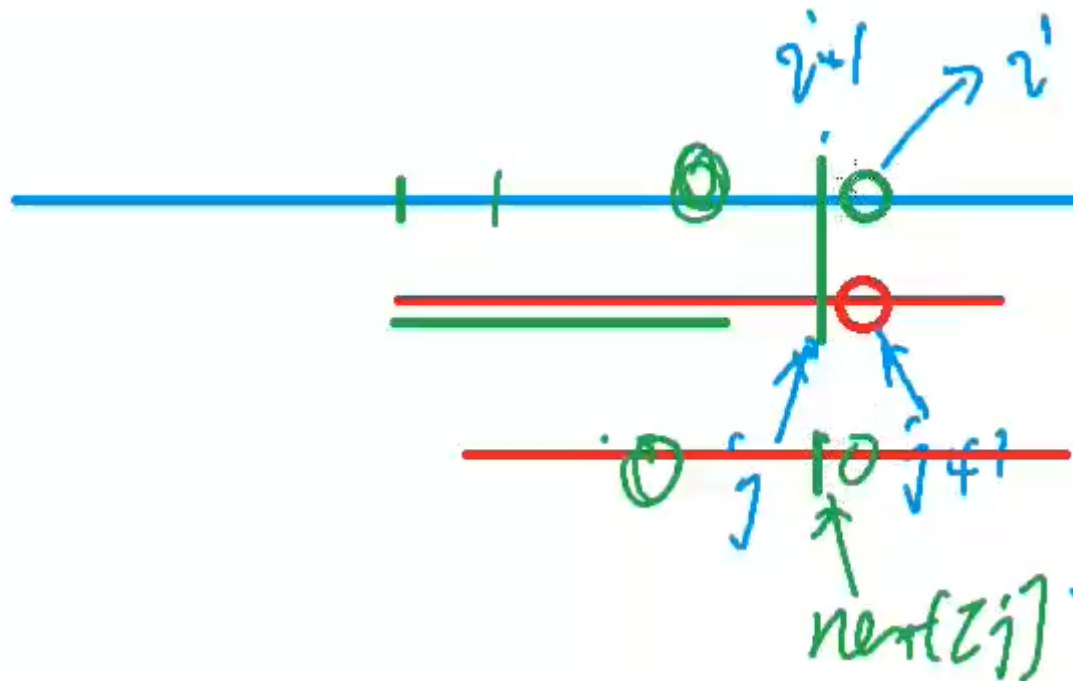
给定一个模式串  $S$ ，以及一个模板串  $P$ ，所有字符串中只包含大小写英文字母以及阿拉伯数字。模板串  $P$  在模式串  $S$  中多次作为子串出现。求出模板串  $P$  在模式串  $S$  中所有出现的位置的起始下标。

### • 思路

先用的字符串哈希，刚开始被卡了，后面问题解决了 帖子在这 [link](#)

$next[i]$  的含义: 前  $i$  个字母构成的字符串中，最长的与前缀相等的后缀长度（非平凡）如  $abaabc$   
 $next[5] = 2$  {% note info %} 为什么最长? 前后缀长度越短，往后移动的就越短 {% endnote %} {% note info %} ps: 平凡的前后缀为字符串本身 {% endnote %}

$i$  是遍历  $s$  的所有字母（从 1 开始）， $j$  是从 0 开始往前走看能不能走通，即  $j = ne[j]$  ( $j$  始终与  $i$  错一位)



从图中可以看到，当  $i$  与  $j+1$  不匹配时， $j$  要回退， $j = ne[j]$  相当于将  $j$  退到它的最长的与前缀相等的后缀长度的地方，其实相对来说，就是模式串往右移动了

### • 代码

- 字符串哈希解法

```

'''
Author: NEFU AB-IN
Date: 2022-02-27 21:04:00
FilePath: \ACM\Acwing\831.py
LastEditTime: 2022-02-27 21:14:38
'''

N = int(1e6 + 1e5 + 100)
h, p = [0] * N, [0] * N
p[0] = 1

P, MOD = 131, 1 << 64

def get(l, r):
    return (h[r] - h[l - 1] * p[r - l + 1] % MOD) % MOD

n_ss = int(input())
ss = input()
ss = " " + ss
res_ss = 0
for i in range(1, n_ss + 1):
    res_ss = (res_ss * P % MOD + ord(ss[i])) % MOD

n_s = int(input())
s = input()
s = " " + s

for i in range(1, n_s + 1):
    h[i] = (h[i - 1] * P % MOD + ord(s[i])) % MOD
for i in range(1, n_ss + 1):
    p[i] = p[i - 1] * P % MOD

for i in range(1, n_s - n_ss + 2):
    if get(i, i + n_ss - 1) == res_ss:
        print(i - 1, end=" ")

```

- \$KMP\$ 复杂度\$O(n + m)\$

```

'''
Author: NEFU AB-IN
Date: 2022-02-28 20:22:41
FilePath: \ACM\Acwing\831.1.py
LastEditTime: 2022-02-28 22:19:15
'''

N = int(1e5 + 10)

```

```
M = int(1e6 + 10)
ne = [0] * N

n = int(input())
p = input()
p = " " + p

m = int(input())
s = input()
s = " " + s

j = 0
for i in range(2, n + 1):
    while j and p[i] != p[j + 1]:
        j = ne[j]
    if p[i] == p[j + 1]:
        j += 1
    ne[i] = j

j = 0
for i in range(1, m + 1):
    while j and s[i] != p[j + 1]:
        j = ne[j]
    if s[i] == p[j + 1]:
        j += 1
    if j == n:
        print(i - n, end=" ")
        j = ne[j]
```