Search pattern P in text T.

$|T| = n$ abacababaacbaba

$|P| = m$ aba

$h_p = hash(p) = hash(\text{"aba"})$

$|P| = m$ $O(m)$

aba ✓
bac
aca
cab

aba ✓
bab
aba ✓
baa
aac

acb
cba
bab
aba ✓

$n \sim 10^5$ $O(n \cdot m)$
$m \sim 10^3 - 10^4$ $O(n+m)$ ✓

Hash values effeciently.

Time to compute hash value of all substrings of length 'm' in T

$(n-m+1)m \longrightarrow n+m$

$O(nm - m^2 + m) \longrightarrow O(n+m)$

$O(nm - m^2 + m + m + n - m + 1) \rightarrow O(n+m+m+n-m+1)$

Time complexity: $\sim O(nm) \longrightarrow O(n+m)$

Fast computation of hash of substrings of a string.

$s = \underline{abaa}\,bcdbab$

$|s| = n$ $\quad i = 0,\ j = 3 \quad 4$

$h(s[i \cdots j]) \longrightarrow$

$h(s[i \cdots j+1])$

Problem:- Find the number of unique substrings in s

$O(n^2)$ of length n

$\sim n \cdot n^2 \log n$ $\quad \sim \underline{n^4}$

$\sim n^3 \log n$

$j-i+1$

$$h(s[i \cdots j+1]) = \left(h(s[i \cdots j]) + \underline{s[j+1]} \times p^{j-i+1}\right) \% \text{ mod}$$

$\underline{O(n^2)} \leftarrow$ compute the hash value of all substrings

$-x-$

— x —

$$hash(s[i \cdots j]) = \sum_{k=i}^{j} s[k] \cdot p^{k-i} \bmod m$$

$$= s[i] \cdot p^0 + s[i+1] \cdot p^1 + \cdots + s[j] \cdot p^{j-i} \bmod m$$

$$hash(s[i \cdots j]) \cdot p^i = \sum_{k=i}^{j} s[k] \cdot p^k \bmod m$$

$$= hash(s[0 \cdots j]) - hash(s[0 \cdots i-1]) \bmod m$$

$$s[i] \cdot p^i + s[i+1] \cdot p^{i+1} \cdots + s[j] \cdot p^j$$

$$\cancel{s[0]} \cdot p^0 + \cancel{s[1]} \cdot p^1 + \cdots \cancel{s[i-1]} \cdot p^{i-1} + \underbrace{s[i] p^i + \cdots + s[j] p^j}$$

$$- \cancel{s[0]} \cdot p^0 + \cancel{\ } \cdots \quad \cancel{s[i-1]} \cdot p^{i-1}$$

$$hash(s[i \cdots j]) \cdot p^i = hash(s[0 \cdots j]) - hash(s[0 \cdots i-1]) \bmod m$$

a b a c d . . b
0 1 2 3 ... n-1

$i=3$, $j=3$
$i=3$, $j=4$

$c * p^0$
$+ d * p^1$
$\vdots$

$h[1 \ldots 1]$
$h[1 \ldots 2]$
$\vdots$
$h[1 \ldots n-1]$

$h[0..0] = h(a)$
$h[0 \ldots 1] = h(ab) = h(a) + b * p^1$
$h[0..2] = h[aba] = h(ab) + a * p^2$
$\vdots$
$h[0 \ldots n-1] = h(0..n-2) + b * p^{n-1}$

```
vll PP(N+1, 1);
ll poly_hash(string &s) {
    ll ans = 0;
    FOR(i, 0, s.length())
        ans = (ans + (s[i]-'a'+1)*PP[i])%MOD;
    return ans;
}


ll number_unique_substrings(string &s) {
    int n = s.length();
    set<ll> ans;
    ll last_hash;
    FOR(i,0,n) {
        last_hash = 0;
        FOR(j,i,n) {
            last_hash = (last_hash+(s[j]-'a'+
1)*PP[j-i])%MOD;
            ans.insert(last_hash);
        }
    }
    return ans.size();
}
```

```
void solve() {
    string s;
    cin >> s;
    cout << number_unique_substrings(s) << endl;
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    cin >> t;
    FOR(i,1,N+1) PP[i] = (PP[i-1]*P)%MOD;
    while(t--) {
        solve();
    }
    return 0;
}
```

https://acmp.ru/asp/do/index.asp?main=task&id_course=2&id_section=18&id_topic=42&id_problem=262

Robin karp algorithm

$|T| = n$ ababbababa $|T| = 10$

$|P| = m$ aba $|P| = 3$

$h(aba) \rightarrow O(m)$

$h(s[i \cdots j])$

$h(s[i+1 \cdots j+1])$

$$h(s[\overbrace{i \cdots j}^{m}]) = s[i] + \boxed{s[i+1] \cdot p + s[i+2] \cdot p^2 + \cdots + s[j] \cdot p^{j-i}}$$

$$h(s[\underbrace{i+1 \cdots j+1}_{m}]) = s[i+1] + s[i+2] \cdot p + \cdots + s[j] \cdot p^{j-i-1} + s[j+1] \cdot p^{j-i}$$

$$h(s[i+1 \cdots j+1]) = \frac{h(s[i \cdots j]) - s[i]}{p} + s[j+1] \cdot p^{j-i}$$

$$= (h(s[i \cdots j]) - s[i]) \cdot p^{-1} + s[j+1] \cdot p^{j-i} \mod m$$

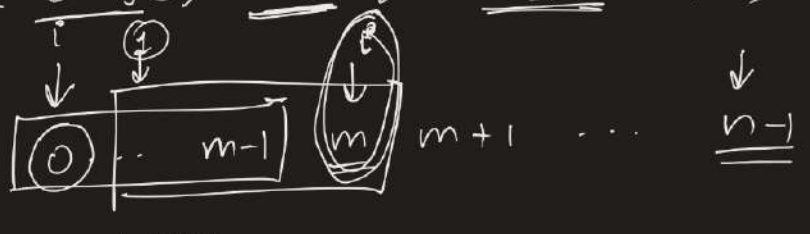
$$p^{-1} = 31^{-1} \mod M = pow(31, M-2, M)$$

$$(h(s[i \cdots j]) - s[i]) \cdot p^{-1} + s[j+1] \cdot p^{m-1} \mod m$$

$$p^{m-1} \to p^{m} \cdot p^{-1}$$

$$= (h(s[i \cdots j]) - s[i]) + s[j+1] \cdot p^{m}) \, p^{-1} \mod m$$

$|T| = n$



remove

$t[i-m]$

$t[i]$

adding

```cpp
#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
#define pb push_back
typedef vector<int> vi;
typedef vector<string> vs;
typedef long long int ll;
typedef unsigned long long int  ull;
typedef vector<ll> vll;
typedef vector<ull> vull;
const int MOD = 1e9+7;
const int P = 31;
const int Pi = 129032259;
void solve() {
    string t,p;
    cin >> t >> p;
    int n = t.length(), m=p.length();
    if(m>n) return;
    ll hash_pattern = 0, hash_text=0, pp=1;
    FOR(i,0,m) {
        hash_pattern = (hash_pattern+(p[i]-'a'+1)*pp)%MOD;
        hash_text = (hash_text+(t[i]-'a'+1)*pp)%MOD;
        pp = (pp*P)%MOD;
    }
    if(hash_pattern == hash_text) cout << "0 ";
    FOR(i,m,n) {
        hash_text = (hash_text-(t[i-m]-'a'+1)+(t[i]-'a'+1)*pp)%MOD;
        hash_text = (hash_text*Pi)%MOD;
        if(hash_pattern == hash_text) cout << i-m+1 << " ";
    }
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    // cin >> t;
    while(t--) {
        solve();
    }
    return 0;
}
```