

Search pattern P in text T .

$|T|=n$ abacababacababa

$|P|=m$ aba

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

$|P|=m$

$O(m)$

→ aba ✓
→ bac ✓
→ aca ✓
→ cab ✓

aba ✓
bab ✓
aba ✓
baa
aac

acb
cba
bab
aba ✓

$n \sim 10^5$

$m \sim 10^3 - 10^4$

$O(n \cdot m)$

$O(n+m)$ ✓

Hash values efficiently.

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

$(n-m+1)m$

$O(nm - m^2 + m)$

→ $n+m$

→ $O(n+m)$

Time complexity:

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

$\sim O(nm)$

→ $O(n+m)$

Fast computation of hash of substrings of a string.

$S = \text{a b a a b c d b a b}$
 $|S| = n$
 $\begin{array}{cc} i & j \\ 0 & 3 \end{array} \textcircled{9}$

$$h(s[\underline{i \dots j}]) \rightarrow h(s[i \dots j+1])$$

Problem:- Find the number of unique substrings in S

$O(n^2)$ of length n

$$\sim n \cdot n^2 \log n \quad \sim \underline{n^4}$$
$$\sim n^3 \log n$$

$$\textcircled{j-i+1}$$

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

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

-x-

-x-

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

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

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

$$= \underline{\text{hash}(s[0 \dots j])} - \text{hash}(s[0 \dots i-1]) \bmod m$$

$$\rightarrow s[i] \cdot p^i + s[i+1] \cdot p^{i+1} + \dots + s[j] \cdot p^j \checkmark$$

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

$$- s[0] \cdot p^0 + \dots - s[i-1] \cdot p^{i-1}$$

$$\text{hash}(\underline{s[i \dots j]}) \cdot \underline{p^i} = \underline{\text{hash}(s[0 \dots j])} - \underline{\text{hash}(s[0 \dots i-1])} \bmod m$$

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

$$\frac{i=3}{i=3} \quad \frac{j=3}{j=4}$$

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

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

$$\begin{aligned}
 h[0 \dots 0] &= h(a) \checkmark \\
 h[0 \dots 1] &= h(ab) = h(a) + b * p^1 \checkmark \\
 h[0 \dots 2] &= h(aba) = h(ab) + a * p^2 \checkmark \\
 &\vdots \\
 h[0 \dots n-1] &= h(0 \dots n-2) + b * p^{n-1} \checkmark
 \end{aligned}$$

```

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;
}

```

Robin karp algorithm

$|T| = n$ a b a b b a b a b a

$|T| = 10$

$|P| = m$ a b a

$|P| = 3$

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

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

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

$$\begin{aligned}
 \checkmark h(s[i \dots j]) &= s[i] + \left[s[i+1] \cdot p + s[i+2] \cdot p^2 + \dots + s[j] \cdot p^{j-i} \right] \\
 \checkmark h(s[i+1 \dots j+1]) &= \underbrace{s[i+1]} + s[i+2] \cdot p + \dots + s[j] \cdot p^{j-i-1} + \underbrace{s[j+1] \cdot p^{j-i}} \\
 h(s[i+1 \dots j+1]) &= \frac{h(s[i \dots j]) - s[i]}{p} + s[j+1] \cdot p^{j-i}
 \end{aligned}$$

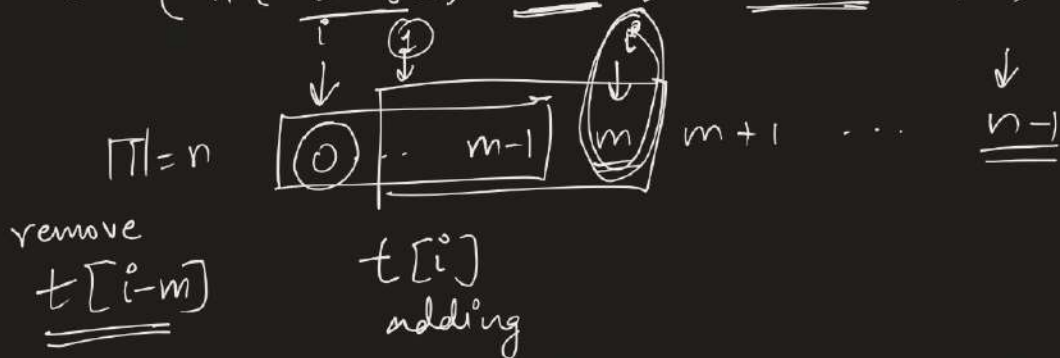
$$= (h(s[i \dots j]) - s[i]) \cdot \underline{p^{-1}} + \underline{s[j+1]} \cdot \underline{p^{j-i}} \pmod m$$

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

$$\Rightarrow (h(s[i \dots j]) - s[i]) \cdot \underline{p^{-1}} + s[j+1] \cdot \underline{p^{m-1}} \pmod m$$

\downarrow
 $\underline{p^m \cdot p^{-1}}$

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




```

#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;
}

```