

# Xử lí xâu

- hash
- z-function
- Trie

# Một số kí hiệu

- $|S|$  độ dài chuỗi  $S$
- $S[i..j]$  đoạn con từ vị trí  $i$  đến  $j$
- Tiền tố là đoạn đầu của một chuỗi
- Hậu tố là đoạn cuối của một chuỗi

# hash

```
1  const int base = 31;
2  const ll MOD = 1e9 + 7;
3
4  ll POW[maxn], hash[maxn];
5  string t;
6
7  void init() {
8      POW[0] = 1;
9
10     for (int i = 1; i <= t.size(); i++)
11         POW[i] = (POW[i - 1] * base) % MOD;
12
13     for (int i = 1; i <= t.size(); i++)
14         hash[i] = (hash[i - 1] * base + t[i - 1] - 'a' + 1) % MOD;
15 }
16
17 ll getHash(int i, int j) {
18     return (hash[j] - hash[i - 1] * POW[j - i + 1] + MOD * MOD) % MOD;
```

# ví dụ

## 1) Tìm xâu (ssf.\*)

Cho xâu T và S, tìm tất cả các vị trí trong T mà S xuất hiện

## 2) Xâu đối xứng dài nhất (lpss.\*)

Cho xâu s, tìm đoạn con đối xứng có độ dài lớn nhất

## 3) So sánh (strcmp.\*)

Cho 2 xâu A, B và q câu hỏi (x, y, u, v) để so sánh A[x..y], B[u..v]

## 4) Xâu con dài nhất bậc k (klss.\*)

Cho xâu s và số nguyên k, tìm đoạn con có độ dài lớn nhất và xuất hiện trong s ít nhất k lần

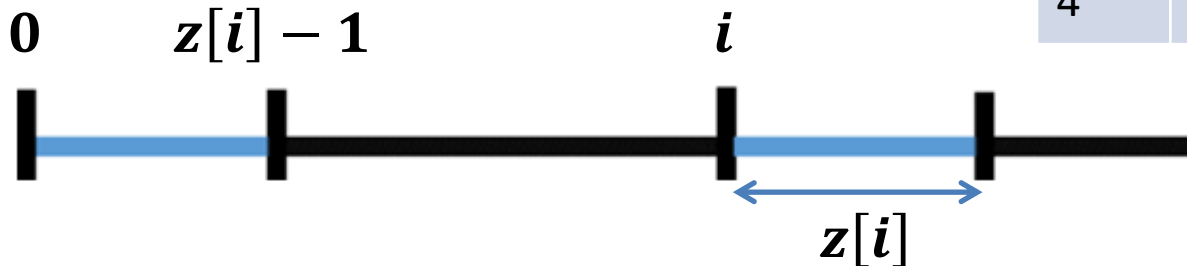
Mã bài	Input	Output	Giới hạn
Tìm xâu (ssf.*)	abab ab	1 3	$ T ,  S  \leq 1e6$
Xâu đối xứng dài nhất (lpss.*)	4 abcb	3	$ S  \leq 1e6$
So sánh (strcmp.*)	5 6 abcde aaddee 3 1 2 1 2 4 5 4 5 4 5 5 6	$\geq <$	$ A ,  B , q \leq 1e6$
Xâu con dài nhất bậc k (klss.*)	5 2 abbba	2	$ S , k \leq 1e5$

# z-function

Xâu  $s$  độ dài  $n$ :  $s[0..n-1]$

Mảng  $z[n]$ , trong đó  $z[i]$  là độ dài tiền tố chung lớn nhất của  $s[0..n-1]$  và  $s[i..n-1]$

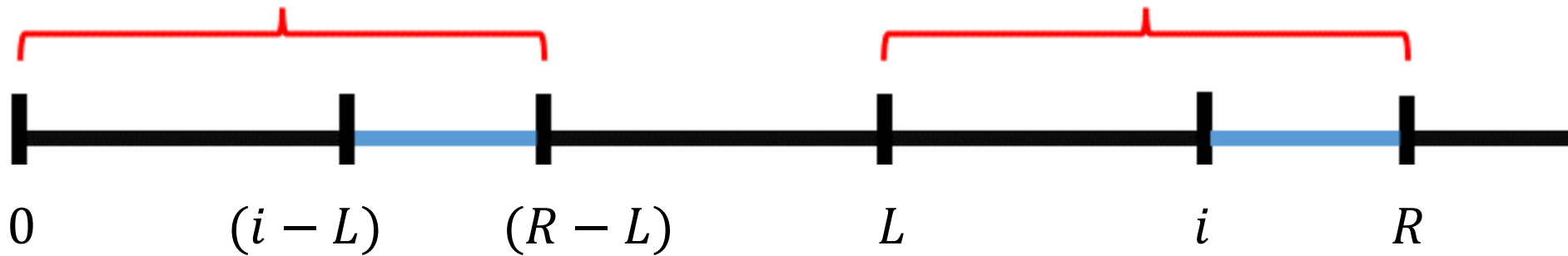
$i$	$s[i..n-1]$	$z[i]$
0	ababa	0
1	baba	0
2	aba	3
3	ba	0
4	a	1



```
1  vector<int> z_function(string s) {  
2      int n = s.length();  
3      vector<int> z(n);  
4      for (int i = 1; i < n; i++)  
5          while (i + z[i] < n && s[z[i]] == s[i + z[i]]) z[i]++;  
6      return z;  
7  }
```

$$R - L + 1$$

$$z[L] = R - L + 1$$



```

1  vector<int> z_function(string s) {
2      int n = s.length();
3      vector<int> z(n);
4      for (int i = 1, L = 0, R = 0; i < n; i++) {
5          if (i <= R)
6              z[i] = min(R - i + 1, z[i - L]);
7          while (i + z[i] < n && s[z[i]] == s[i + z[i]]) z[i]++;
8          if (i + z[i] - 1 > R) {
9              L = i;
10             R = i + z[i] - 1;
11         }
12     }
13     return z;
14 }

```

# ví dụ

## **1) Tìm xâu (ssf.\*)**

Cho xâu T và S, tìm tất cả các vị trí trong T mà S xuất hiện

## **2) Xâu con khác nhau (dss.\*)**

Cho xâu S, đếm số lượng đoạn con khác nhau.

**So sánh với cách dùng hash?**



Mã bài	Input	Output	Giới hạn
<b>Tìm xâu (ssf.*)</b>	abab ab	1 3	$ T ,  S  \leq 1e6$
<b>Xâu con khác nhau (dss.*)</b>	0101	7	$ S  \leq 5e3$

# trie

```
1  const int LIM = ...; //LIM = tổng số kí tự của tập xâu
2  struct node{
3      int child[26];
4      int cnt;
5  } trie[LIM];
6  int nNode = 0;
7
8  void add_string(string s) {
9      int cur = 0;
10     for (auto ch : s) {
11         int c = ch - 'a';
12         if (trie[cur].child[c] == -1) {
13             nNode++;
14             memset(trie[nNode].child, -1, sizeof(trie[nNode].child));
15             trie[nNode].cnt = 0;
16
17             trie[cur].child[c] = nNode;
18         }
19         cur = trie[cur].child[c];
20     }
21
22     trie[cur].cnt++;
23 }
```