Хеши

$\langle m, x\rangle$ $m \in P$ $x \in \mathbb{Z}_m$

$h_{m,x}(s) = \sum_{i=0}^{|s|-1} s_i \cdot x^{|s|-i-1} \bmod m$

1). строчки случайные

2). Генер. случ. сами: x – выбир. случайно

1). s, t – случ. мн-гл. $n = \max(|s|, |t|)$

$\Pr_{s,t}[h_{m,x}(s) = h_{m,x}(t)] = \Pr_{s,t}[\sum_{i=0}^{n}(s_i - t_i)x^{n-i-1} \bmod m = 0] =$ (случ. мн.-гл. r)

$= \Pr_r[r(x) \vdots m] = \frac{1}{m}$ $(\equiv_m 0)$

$r(x) = x(\ldots) + r_{n-1}$ ← св. гл.

$\Rightarrow r(x)$ р/м распр. на $[0, m-1]$

k сравнений $\Rightarrow \Pr[\exists$ коллизия$] = \Pr[$колл$_1 \vee$ колл$_2 \vee \ldots \vee$ колл$_k]$

$\le \sum_{i=1}^{k} \Pr[$колл$_i] = \frac{k}{m}$

2). s, t – фикс. $|s|, |t| \le n$
$s \ne t$ $m \in P$ – фикс. $x \leftarrow U_{\{0,\ldots,m-1\}}$

$\Pr_x[\sum(s_i - t_i) \cdot x^{n-i-1} \bmod m = 0] = \Pr_x[r(x) \equiv_m 0] = \frac{\#\text{корней}}{m} \le \frac{n}{m}$ ($r_i = s_i - t_i$; r – фикс. r = s – t)

$\deg r \le n-1 \Rightarrow$ корней $\le n-1$

(r – не нулевой мн-гл., т.к. $s \ne t$)

k сравнений

$\Pr[\exists \ge 1$ коллизия$] \le \frac{kn}{m}$ {на прак. $\frac{k}{m}$}

$m \sim 10^{18}$

$(h[r] - h[l] \times \deg[r - l]) \bmod m$ ($< m$, $< m$)

1). mul(ll a, ll b, ll m): a*b mod m

→1.1). (mul(a, b/2, m)*2 mod m + a) mod m, b%2==1 } log b

→1.2). long double O(1)

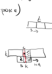
2). $m = p_1 p_2$ $p_1 = 10^9+7$ $p_2 = 10^9+9$
(r_1, r_2) +, *, -

Алг. Бойера-Мура

$O(|t| \cdot |s|)$ в среднем $O(\frac{|t|}{|s|})$

(ищем подстроку в строке)

```
for (p=0; p <= |t|-|s|; ++p)
    for (k=|s|-1; k>=0; --k)
        if t[p+k] != s[k]:
            break
    if k == -1
        return true
```



abac

a: 0,2
b: 1
c: 3

```
for i=0..|s|-1
    pos[s[i]].push_back(i)
for (p=0; p <= |t|-|s|; p+=dp)
    for (k=|s|-1; k>=0; --k)
        if s[k] != t[p+k]:
            break
    if k<0:
        return true
    auto &v = pos[t[p+k]]      // символ c
    for (i=v.size()-1; i>=0 && v[i]>=k; --i);
    if i<0:
        dp = k+1
    else:
        dp = k - v[i]
```

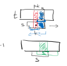
1). правило плох. симв.

2). правило хор. суф.

$z[i]$ – z-функц. для $\bar{s}$

совпал. суфф. u

j_min:
$z[j] \ge |u| = |s| - k - 1$
$z[j] = |u|$!



```
for j=|s|-1; j>=0; --j
    shift[z[j]] = j
```

$shift[x] = \min\{j \mid z[j] = x\}$

$p += \max(dp, shift[|s| - k - 1])$

// shift[x] = ∅ → ?

~~Предподсчёт: T = O(|s|) M = O(|s|)~~

Суффиксный массив

$S = s_0 s_1 \ldots s_{n-1}$

$sa = [sa_0, sa_1, \ldots, sa_{n-1}]$

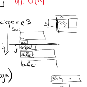
суффиксы:
0: $s_0 s_1 \ldots s_{n-1}$
1: $s_1 s_2 \ldots s_{n-1}$
⋮
n-1: s_{n-1}
} n строк

$s[sa_i:] < s[sa_j:] \Leftrightarrow i < j$

1). $O(n^2 \log n)$ – наивно
2). $O(n \log^2 n)$ – хешами
2.5). $O(n^2)$
3). $O(n \log n)$
4). $O(n)$

Применение: поиск подстроки x в строке s
нужно найти $\min_j: s[sa_j:] \ge x$

1). бинпоиск + сравнение за O(n)
$O(\log|s| * |x|)$



2). бинпоиск + сравнение за O(log n)
$O(\log|s| \times \log|x| + |x|)$

3). $O(\log|s| + |x|)$

Будем считать, что $|\Sigma| \le n = |s|$. Упр. Что если нет?

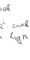
$O(n^2)$:

$n \times (n + |\Sigma|) = O(n^2)$

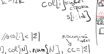
• Сортируем не суффиксы, а циклич. сдвиги

ababb∅

суфф.:	sa:
ababb	ababb.
babb	abb.
abb	b.
bb	babb.
b	bb.
ababb∅	ababb∅
babb∅a	abb∅ab.
abb∅ab	b∅ababb.
bb∅aba	babb∅a.
b∅abab	bb∅aba.



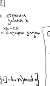
Упр. s: сорт. суф. ≠ сорт. цикл. сдвигов



На шаге i сорт. по 2^i симв.
⇒ после шага log n

Переход k→2k

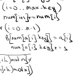
p[i] = какая строка длины k лежит на i месте в ...
col[i] = цвет строки s[i : i+k]



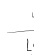
col[p0] = 0 = col[p1] = col[p2] = ...
col[p5] = 1 = col[p6]

n штук

key = col[(p[i] - k + n) mod n]
(col[...], col[p[i]])



```
//0<=s[i]<|Σ|                       последний цвет
int p[N], col[N], num[N], cc=|Σ|
for (k=0; k<n; k=k?k*2:1)   //отсорт. строки длины k
                             //в col хр-ся цвета подстрок длины k
    num ← нули
    for (i=0..n-1)
        num[col[i]+1] += 1
    for (i=0..cc)
        num[i+1] += num[i]
    for (i=0..n-1): key = col[(p[i]-k+n) mod n]
        p2[num[key]] = (p[i]-k+n) mod n
        num[key] += 1
    cc=0
    for (i=0..n-1)
        if (i>0 and (col[p2[i]] != col[p2[i-1]] or col[(p2[i]+k) mod n] !=
                    != col[(p2[i-1]+k) mod n])
            cc += 1
        col2[p2[i]] = cc
    p=p2, col=col2
for i=0..n-1 { p[i]=i; col[i]=s[i]; }
```



```
a – массив → b
a[i].key
for (i=0..n-1)
    num[a[i].key] += 1
for (i=0..max_key)
    num[i+1] += num[i]
for (i=0..n-1)
    b[num[a[i].key]] = a[i]
    num[a[i].key] += 1
```

j-k j
abac
k k=2
ab
ba
ac
ca

p:		i	-2%4	
0	ab	0	2 ac	ab
2	ac	1	0 ab	ac
1	ba	2	3 ca	ba
3	ca	3	1 ba	ca

p[i]-k p[i]
s

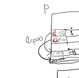
k=0 → k=1 → k=2 → k=4 → k=8...

s[p[i] : p[i]+k] – на i-м месте при сорт. подстрок длины k

4). O(n) – Карккайнен, Сандерс

LCP – largest common prefix

$LCP[i, j] = \max_k : s[i : i+k] == s[j : j+k]$



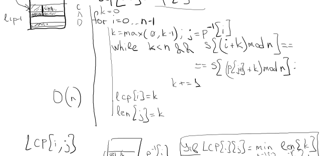
Будем считать, что все цикл. сдвиги различны

aa → aa, aa
aa∅ → a∅, aa∅

1). O(n log n) – хешами
2). O(n)

Алг. Касаи-Ли-Аримуры-Арикавы-Парка

$p[i]$ | $p^{-1}[i] = j : p[j] = i$



p[j] = i
p[j+1]
p[j'] = i+1

A<B ⇒ C<D

lcp[i] > 0 ⇒

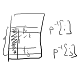
$lcp[i+1] \ge lcp[i] - 1$

i < j < k

```
k=0
for i=0..n-1
    k=max(0,k-1); j=p^-1[i]
    while k<n && s[(i+k) mod n] ==
                 == s[(p[j+1]+k) mod n]:
        k += 1
    lcp[i]=k
    len[j]=k
```

O(n)

LCP[i, j]



Упр. $LCP[i][j] = \min_{k = p^{-1}[i] .. p^{-1}[j]-1} len[k]$ → O(1) с ФКБ.