Суффиксный массив

- построили за $O(n\log n)$
- LCP за $O(n)$

s

$O(|s| + \log|t| \cdot |s|)$

$O(|s| + \log|t| \cdot \log|s|)$

Поиск подстроки s в тексте t за время $O(|s| + \log|t|)$?

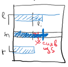
// $lcp(s, i) = lcp(s, t[sa[i]:])$

$x = \max \begin{cases} \min(l_i, lcp(l,m)) \\ \min(r_i, lcp(r,m)) \end{cases}$

можем посчитать за $O(1)$

$l_i = lcp(s, l)$

$r_i = lcp(s, r)$



Утв $lcp(s,m) \geq x$.

Код: $lcp(s,m) = x +$ наивный досчёт

Пусть что-то досчитали на i-й итерации бинпоиска

$l_i = lcp(s, l) \geq r_i = lcp(s, r_i)$ $\quad z_i = \max(l_i, r_i)$

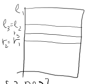
1). $lcp(s,m) < z_i \Rightarrow x = lcp(s,m)$ $\quad m > s \Rightarrow r' = m$
$z_{i+1} = z_i$ $\quad r_{i+1} = lcp(s,m)$ $l_{i+1} = l_i$

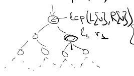
2) $lcp(s,m) \geq z_i$ $\quad \begin{cases} x = z_i \\ z_{i+1} = lcp(s,m) \end{cases}$ $\quad (lcp(s,m) - x) = $ #операций
Δz

$\sum$ #операций $= \sum \Delta z = z_{final} - z_{start} \leq |s|$

Time $= O(|s| + \log t)$

Как находить $lcp(l,m)$? $lcp(r,m)$?



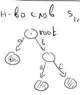


$[L[v], R[v]]$ – отрезок sa, за кот. отвеч. вершина v д.о.

Бор (trie)

мн-во слов $s_1, \ldots, s_n$

a, ba, bc



1). массив $next[v][|\Sigma|] = u$
спуск: $O(|s|)$
память: $O(V \cdot |\Sigma|)$

2). list/vector
спуск: $O(|s| \cdot |\Sigma|)$
память: $O(V)$

3). map
3.1). TreeMap $O(|s| \cdot \log|\Sigma|)$ $O(V)$
3.2). HashMap $O(|s|)$ const! $O(V)$

4). SplayMap $O(|s| + \log|S|)$ $O(V)$

– отсортировать строки
TreeMap: $O(S \log|\Sigma|)$ $\quad$ SplayMap: $O(S + n\log S)$
$S = \sum s_i$ $\quad$ n = #строк

text

$\{s_1, \ldots, s_n\}$
$\exists? s_i : s_i \in text$

Поиск словарных слов в тексте

① text

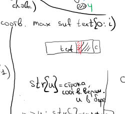
i

for i=0..|t|-1
check(t[i:]) $\quad$ // $\min(|t| - i, \max_{1 \leq j \leq n} |s_j|)$

$O(|t| \cdot \max_{j \in [n]} |s_j|)$

② text c

содержится в боре (⟺ является префиксом какого-то словарного слова)



xabacc,c,z
v[0] = 1 $\quad$ v[5] = 4
v[1] = 1 $\quad$ v[6] = 5
v[2] = 1 $\quad$ v[7] = 1 = root
v[3] = 2
v[4] = 3

v[i] = вершина в боре, соотв. max suf text[0:i)

fake
suf[root] = fake
next[fake][c] = root
$\forall c \in \Sigma$

```
v[0]=root; p=root
for i=0..|t|-1: c=t[i]
    while(next[p][c]==-1)
        p=suf[p]
    p=next[p][c]
    v[i+1]=p
```

str[u] = строка, соотв. верш. u в боре

p→u: str[u] – суффикс str[p], u≠p
str[u] → max

suf[u] = r ∈ бор: (u – верш. бора)
суф. ссылка
1). r ≠ u
2). str[r] – суф. str[u] 3) str[r] → max

v[i]

for i=0..|t|-1
used[v[i]] = true

bfs от листьев $\forall u$: if used[u]: used[suf[u]] = true

Подсчёт суффиксных ссылок

① BFS от корня

```
q=[root]
while(!q.empty()){
    v=q.front(), q.pop_front()
    c=parent_char[v]    //символ p[v]->v
    z=suf[p[v]]
    while next[z][c]==-1:
        z=suf[z]
    suf[v]=next[z][c]
}
```

str[v]
str[p[v]]

Это не работает за $O(V)$.
Это работает за $O(S)$, $S = \sum|s_i|$

{a…ab, a…ac, a…ad} – n слов
$\Omega(n^2)$

Для вершин пути, соотв. строке s_i работаем за $\leq 2|s_i|$
$\Rightarrow$ Time $= O(\sum 2|s_i|) = O(S)$
(считая, что у нас массив / HashMap)

Общее время работы: Build($\{s_1, \ldots, s_n\}$) + BuildSufLinks + цикл по text
HashMap: $O(S)$ $\quad$ $O(S)$ $\quad$ $O(|t|)$
Mem: $O(S)$

Ахо, Корасик

② Достроение до ДКА

Мемоизация go[v][c] suf[v]

```
go(v,c):
    if next[v][c] != -1
        return next[v][c]
    else:
        return go(suf(v),c)

suf(v):
return go(suf(p[v]), parent_char[v]);

suf(root) = fake
```

next[fake][c] = root $\forall c \in \Sigma$

for i=0..|t|-1
v[i]=go(v[i], t[i])

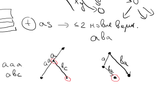
Time $O(S) + O(V \cdot |\Sigma|) + O(|t|)$
Mem $O(V \cdot |\Sigma|)$

Суффиксное дерево – бор всех суффиксов строки

Сжатый бор

text
s

st



Утв сжатое суф. дерево содержит $\leq 2|s|$ вершин.
строки s

Пусть с.с.д. a s $\oplus$ as $\rightarrow \leq 2$ новые верш.

aba

Укконен: $O(|s|)$

suf[v] имеет длину ровно на 1 меньше

aaa
abc