

Строки

Ⓘ Найти подстроку в строке in

Текст t
подстрока s

t : a b c a b c d
 s : ab

Тривиальное решение:

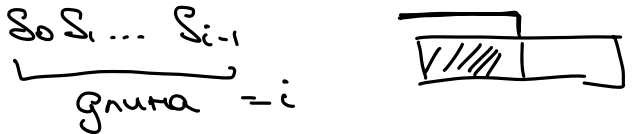
```
for i = 0... t.size()
{
    flag = true
    for j = 0... s.size()
    {
        если  $t[i+j] \neq s[j]$ :
             $\hookrightarrow$  flag = false
    }
    если flag = true:
         $\hookrightarrow$  вернуть (i)
}
```

Асимптотика:

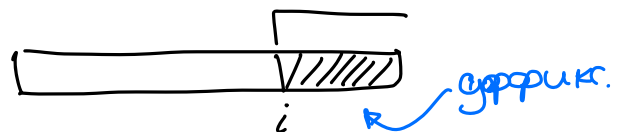
$O(|s| \cdot |t|)$

Ⓜ Префикс - функция (π)

Опр Префиксом строки s длины i наз-ся подстрока $s_0 s_1 \dots s_{i-1}$

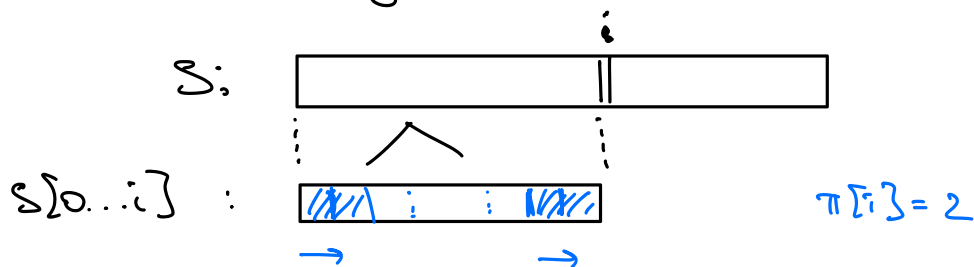


Опр Суффиксом строки s длины i наз-ся подстрока $s_{-i} \dots s_{-1}$



Опр Совместный префикс (суффикс) - префикс (суффикс), не совпадающий со всей строкой.

Опр Префикс-функцией (π/p) строки s наз-во массив π , где $\pi[i]$ - длина наибольшего собственного префикса подстроки $s[0...i]$, совпадающий с суффиксом этой подстроки.



Пример:

$\pi[0] = 0$

$s: \quad a \ b \ a \ b \ c \ a \ b$

$\pi: \quad 0 \ 0 \ 1 \ 2 \ 0 \ 1 \ 2$

Вычисление:

```

prefix-function (string s)
  p = [0 ... 0] // размер s.size()
  N = s.size()
  for i = 1 ... N-1 : // префикс
    for k = 1 ... i : // длина
      // нр. нр.
      если s[0... k] = s[i-k+1... i]
        p[i] = k
  
```

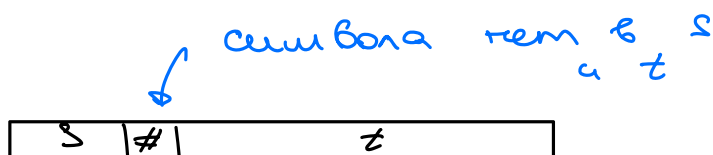
Асимптотика: $O(N^3)$

Ⓜ Для чего это нужно? (Алгоритм Кнута - Морриса - Пратта)

t - текст

s - строка

длина: $|s| + |t| + 1$



1. Вычисление π (префикс-функция)
2. если $\pi[i] = |s| \Rightarrow$ нашли подстроку

Асимптотика: $\underline{0}$ ($|s| + |t|$)

IV) Быстрое вычисление π

ПРИМЕР:

①

\mathcal{S} :	a	a	a	a	a
π :	0	1	2	3	4

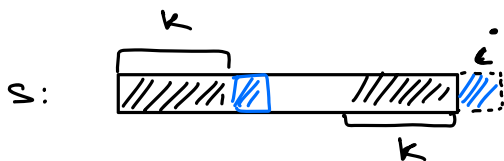
③ $\sigma: a b a b c a$
 $\pi: 0 0 1 2 0 1$

② S: a b c
T: 0 0 0

Zusatz:

$\pi[i+1]$ should $\pi[i]$
be be done 1.

ଦ - ଶବ୍ଦ



$$k = \pi \{i\}$$

$$S[i+1] = S[k+1]$$

$$\hookrightarrow \pi [i+1] = k+1$$

не равно \Rightarrow

⇒ ТОЖО УМЯВШЫ

Realizasyon:

prefix-function (string s)

$p[...] = [a...0]$ // size()

$$N = \text{s. Size()}$$

for $i = 2 \dots N-1$

$$k = p[i-1]$$

while $s[i] \neq s[k]$ and $k > 0$:

$$k = p[k-s]$$

ecnu $s[i] = s[k]$

$$p(i) = k + 1$$

Асимптота:

10 (2)

\Rightarrow Бн Бодь

Answer

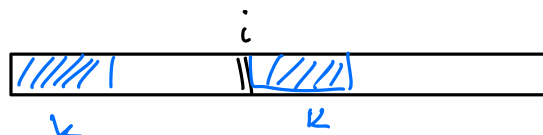
кран - тору

Tippen:

	0	1	2	3	4	5
Σ :	a	b	a	b	a	a
π :	0	0	1	2	3	1

⑤ z-функция.

Опр z-функцией наз-ет массив z, такой что $z[i] = \text{длина максимального префикса подстроки } z[0...i-1], \text{ совпадающего с префиксом } z[i...N]$



a	b	c	d	a	b	c
0	1	2	3	4	5	6

$z[4] = 3$

Алгоритм:

z-function (string s):

$N = s.size()$

$z = \{0... \}$ // размер N

for $i = 1..N-1$

while $i + z[i] < n$ and $s[z[i]] = s[i + z[i]]$

↳ $z[i]++$

Асимптотика:

$O(N^2)$

Магическая

Алб xmp

z | # | s

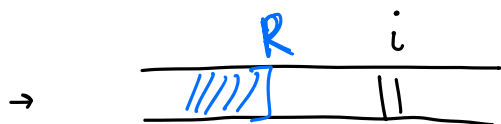
⑥ Быстрая вычисления z-function

a) z-блок



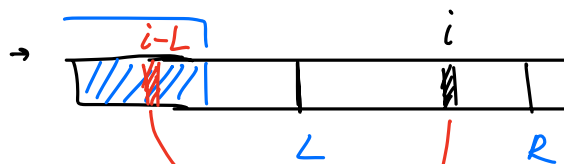
z-блок = префиксу (самый правый из найденных)

б)

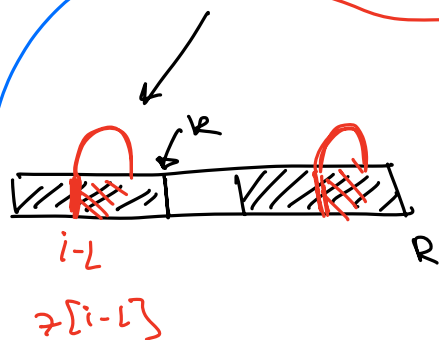


⇒ не можем использовать z-блок не зная

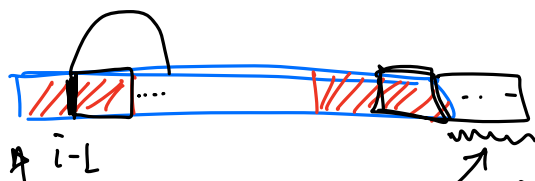
$z[i] = 0$



Роль: $s[i-L] = s[i]$
 и группа
 z-функции



$k = R - i + 1$
 если $z[i-L] < k$
 $\Rightarrow z[i] = z[i-L]$



сравниваем
 с текущим. строку s

Реализация:

z-function (string s)

$z = [0 \dots]$

$N = s.size()$

$L = R = 0$

for $i = 1 \dots N-1$

если $i \leq R$

$z[i] = \min(z[i-L], R-i+1)$

сравниваем
 с группой
 z-функции

расширяем
 z-блок

while $(i+z[i] < N \text{ and } s[i+z[i]] = s[i+z[i]])$
 $z[i]++$

если $i+z[i] - 1 > R$

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

$L = i$

сдвигаем
 z-блок

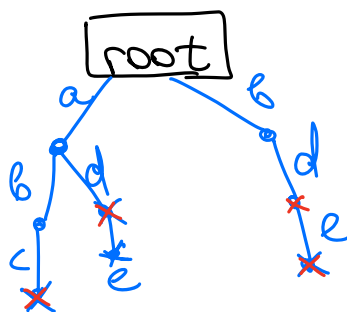
Асимптотика: $O(N)$ - те всего проходим R

VI)

Продуктовое дерево (Bor)

структура данных графа хранилище симв

abc
ad
bde
bd



Node {

Node* to[] = {nullptr...}

bool terminal = false

}

узел корня
символ

add_string (string s) {

runner = root

for i = 0... s.size()

char = s[i] - 'a'

if runner -> to[char] == nullptr:

↳ runner -> to[char] = new Node

runner = runner -> to[char]

runner -> terminal = true

find, erase