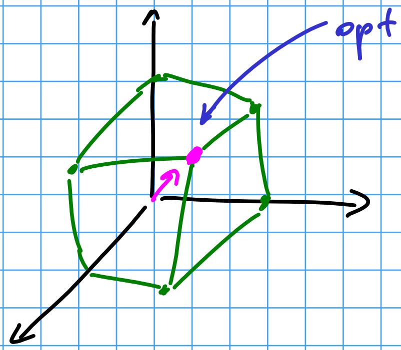
Линейное программирование

Линейная программа

$$\begin{cases} x_1 + x_2 \le 100 \\ 2x_1 + x_3 \le 200 \\ x_2 + x_3 = 200 \end{cases}$$

$$x_1, x_2, x_3 \ge 0$$

$$15x_1 + 20x_2 + 5x_3 \to \max$$



Ссылки:

- Дасгупта
- Курс Макса Бабенко на сайте CS клуба

Поиск подстрок

Обозначения:

P – строка длины m

$P[1..m]$ – вся строка

$P[i..j]$ – подстрока $[i, j]$

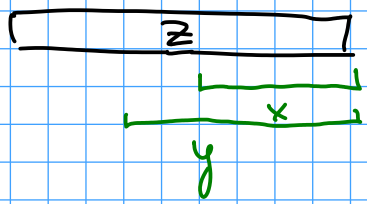
x, y – строки

$x \sqsubset y$ – x – префикс y

$x \sqsupset y$ – x – суффикс y

Лемма:

$$x \sqsupset z,\ y \sqsupset z,\ |x| \le |y| \Rightarrow x \sqsupset y.$$



ч.т.д.

Задача о поиске подстроки:

Вход: $T[1..n]$ – текст

$P[1..m]$ – шаблон, искомая подстрока

Выход: $\{i : T[i..(i+m-1)] = P[1..m]\}$

Наивный алгоритм

```
Search (T, P):
  for i = 1 to (n - m + 1)
      if T[i..(i+m-1)] = P        O(m)
          return i
  return ∅
```

$$O(m \cdot (n - m + 1))$$

$$O(m \cdot n)$$

Алгоритм Рабина-Карпа

Для $T[1..n]$ — $P[1..m]$

вычисляем $t_1, t_2, \ldots, t_{n-m+1}$ — хеши подстрок $T[1\ldots m], \ldots\ldots T[n-m+1\ldots n]$

Для произвольных хешей $O((n-m+1)\cdot m)$

Для полиномиальных хешей

$t_s \longrightarrow t_{s+1}$ за $O(1)$ по модулю простого числа

$$t_s = T[s]\cdot q^{m-1} + T[s+1]q^{m-2} + \ldots + T[s+m-1]$$

$$t_{s+1} = T[s+1]q^{m-1} + T[s+2]q^{m-2} + \ldots + T[s+m]$$

$$t_{s+1} = (t_s - T[s]\cdot q^{m-1})q + T[s+m]$$

Rabin-Karp-Search (T, P)

$p = P[1]q^{m-1} + P[2]q^{m-2} \ldots + P[m]$

$t_1 = T[1]q^{m-1} + \ldots \quad T[m]$

for $i = 1$ to $n-m+1$

  if $t_i = p$

    if $T[i..i+m-1) = P$:

      yield i

  if $i \neq n-m+1$

    $t_{i+1} = (t_i - T[i]q^{m-1})q + T[i+m]$

Пусть вероятность коллизии $< \frac{1}{n}$ (модуль $> n$)

$$O(n + \underbrace{v}_{\text{вхождений}}\cdot m + n\cdot\tfrac{1}{n}\cdot m) = O(n+m+v\cdot m)$$

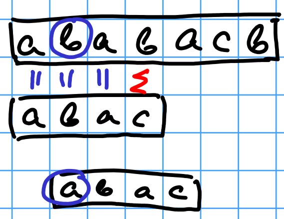
$= O(n+m)$, если $v = O(1)$

Алгоритм Кнута-Морриса-Пратта.

T   ababacb

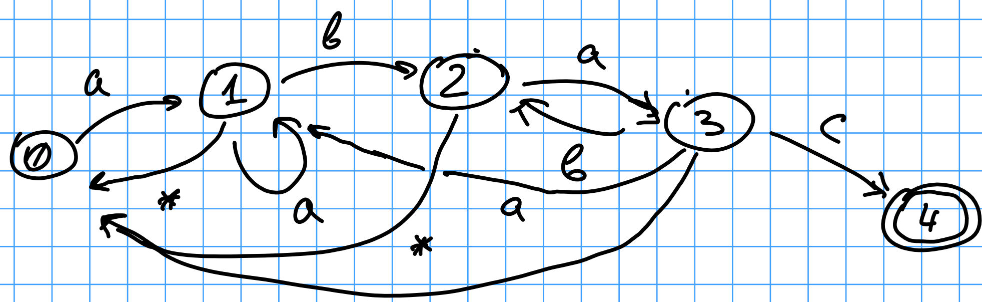
P   abac

abcd



Как использовать эту информацию?

– Конечный автомат



$\Rightarrow$ Поиск за $O(n)$

Построение автомата: $\Omega(m \cdot |\Sigma|)$

$$O(n + m \cdot |\Sigma|)$$

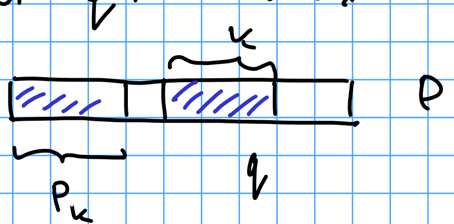
– Префикс функция

P – шаблон

P_i – префикс длины i $P[1..i]$

$\equiv$ $\pi : \{1 \ldots m\} \to \{0 \ldots m-1\}$

$$\pi(q) = \max k < q : P_k \sqsupset P_q$$



abcdefabcf

$\pi[9] = 3$ $P_3 \sqsupset P_9$

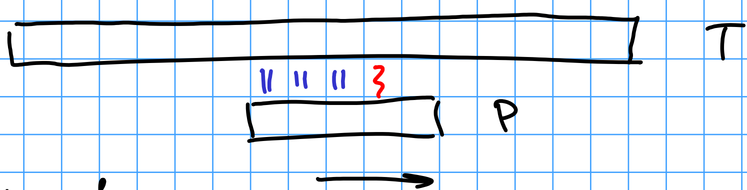
$\pi[3] = 0$

a b a b a b c

π 0 0 1 2 3 4 0

KMP - Search (T, P)

```
π ← Prefix Function (P)
k = 0      // # совпавших символов
for i = 1 to n
1 [   while k > 0 and P[k+1] ≠ T[i]
          k = π[k]
2 [   if P[k+1] = T[i]
          k = k + 1
      if k = m
          yeild i - k + 1
          k = π[k]
```



В блоке (1) $k \downarrow$

В блоке (2) $k \uparrow$ на 1

Сколько итераций (1)? . . . . . a b c d e f a b c d . . . . .

a b c d e f a b c f

Не больше, чем (2) a b c d e f a b c f

$O(n)$

(можно ∡ k как потенциал)

Prefix-Function (P[1..m])

```
∀i π[i] = 0
   k = 0
   for i = 2 to m
       while k > 0 and P[k+1] ≠ P[i]
           k = π[k]
       if P[k+1] = P[i]
           π[i] = k+1
           k = k+1
```

$O(m)$

Итого: $\underbrace{O(m)}_{\text{построение}} + \underbrace{O(n)}_{\text{поиск}} = O(m+n)$