

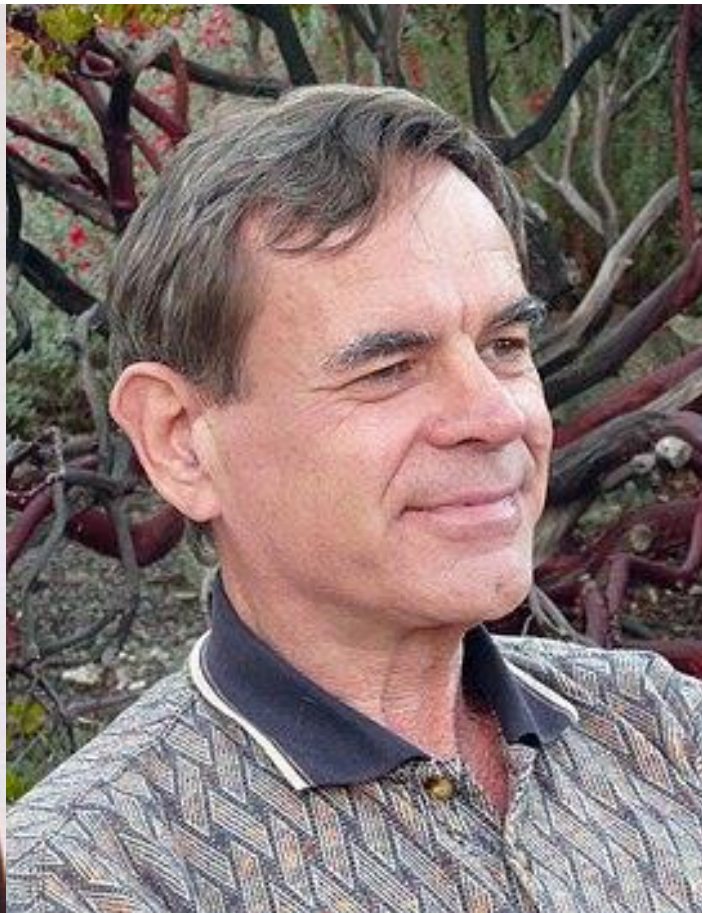
# Алгоритм Кнута-Морриса- Пратта

Выполнил: Закарлюка Иван Владимирович, 23.Б10-ММ

# Разработка



Д.Кнут



В.Пратт



Д.Моррис

# Использование

- Алгоритм КМП используется для поиска вхождений одной строки во вторую
  - Используется для поиска по файлу (например, в текстовых редакторах)
  - Используется в биоинформатике для поиска генетических последовательностей в больших наборах данных

# Префикс функция

$P[i]$	a	b	a	b	a	b	a	b	c	a
$\pi[i]$	0	0	1	2	3	4	5	6	0	1

# Наивная реализация

```
vector<int> slow_prefix(string s) {  
    int n = (int) s.size();  
    vector<int> p(n, 0);  
    for (int i = 1; i < n; ++i)  
        for (int len = 1; len <= i; ++len)  
            if (s.substr(0, len) == s.substr(i - len + 1, len)) {  
                p[i] = len;  
            }  
    return p;  
}
```

# Оптимизации

---

- 1)  $\pi_{i+1}$  не больше, чем  $\pi_i + 1$
- 2) Вычислять значение через предыдущие

Высчитывание значения через  
предыдущие

- Посчитаны значения для  
строки *aabaataaba*

$$\pi = 01012012345$$

- 2 случая:
  - Следующий символ =  $t$
  - Следующий символ  $\neq t$  (пусть будет  $b$ )

*aabaataaabaab*

$\pi = 01012012345x$



## Реализация

```
vector<int> prefix_function(string s) {  
    int n = (int) s.size();  
    vector<int> p(n, 0);  
    for (int i = 1; i < n; ++i) {  
        int len = p[i - 1];  
        while (len > 0 && s[len] != s[i]) {  
            len = p[len - 1];  
        }  
        if (s[len] == s[i]) {  
            p[i] = len + 1;  
        }  
    }  
    return p;  
}
```

# Асимптотика

- Время -  $O(n + m)$
- Память -  $O(n + m)$

Где  $n$  и  $m$  - длины строк

Строка длины  $n$ , которую ищем

  
a b a # b a b a c

Символ, которого нет в строках  
(разделитель)

  
Строка длины  $m$ , в которой ищем

- Если  $\pi_i = n$ , то строка, которую мы ищем, входит в строку, в которой мы ищем на полуинтервале  $[i - n + 1 : i + 1)$

# Асимптотика

- Такая же, как для построения префикс функции

## Замечание

- Вместо префикс функции можно использовать зет-функцию

# Пример работы

```
mujik#kupil mujik shlyapu,a ona emu kak raz  
0000000000000123450000000000000000001200000000  
mujik - подстрока с 6 символ по 11
```

```
aba#abacabacabaaba  
001012301230123123
```

aba - подстрока с 0 символ по 3

aba - подстрока с 4 символ по 7

aba - подстрока с 8 символ по 11

aba - подстрока с 11 символ по 14

# ИСТОЧНИКИ

- [https://neerc.ifmo.ru/wiki/index.php?title=%D0%A4%D0%B0%D0%B9%D0%BB:Kmp\\_pict2.png](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A4%D0%B0%D0%B9%D0%BB:Kmp_pict2.png)
- <https://ru.wikipedia.org/wiki/%D0%9A%D0%BD%D1%83%D1%82,%D0%94%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4%D0%AD%D1%80%D0%B2%D0%B8%D0%BD>
- <https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B0%D1%82%D1%82,%D0%92%D0%BE%D0%BD%D0%A0%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4>
- <https://s3d.cmu.edu/people/affiliated-fac/morris-james.html>
- <http://algcourse.cs.msu.su/wp-content/uploads/2010/09/Lecture17-2018.pdf>
- <https://ru.algorithmica.org/cs/string-searching/z-function/>
- <https://ru.algorithmica.org/cs/string-searching/prefix-function/>