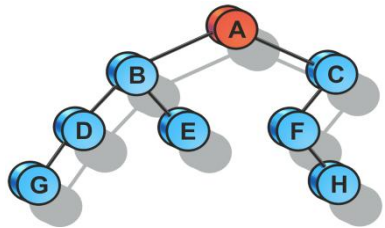


Алгоритмы программирования и структуры данных

Введение в поиск подстрок

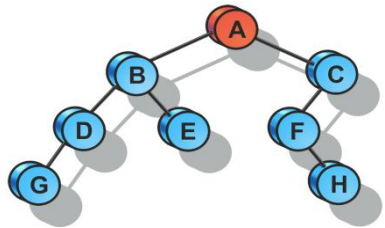
---

## Поиск подстрок (часть 2). Алгоритм Рабина-Карпа



## На чем мы остановились

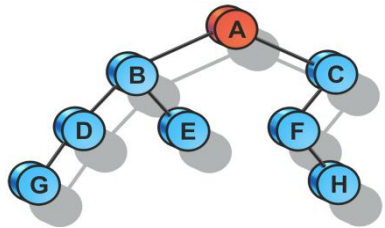
```
find(s, t: String):  
    for i = 0 .. s.length - t.length:  
        ok = True  
        for j = 0 .. t.length - 1:  
            if s[i + j] != t[j]:  
                ok = False  
                break  
        if ok:  
            return i  
    return -1
```



## Хеш-функция

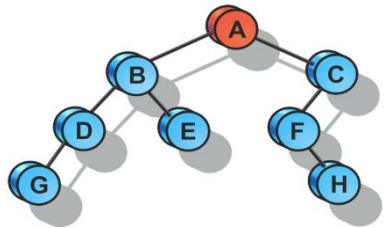
$$\text{hash}(S) \rightarrow [0..M-1]$$

$$\text{hash}(X) \neq \text{hash}(Y) \Rightarrow X \neq Y$$



## Использование в поиске подстроки

```
find(s, t: String):  
    for i = 0 .. s.length - t.length:  
        if hash(s[i : i + t.length]) != hash(t):  
            continue  
        ok = True  
        for j = 0 .. t.length - 1:  
            if s[i + j] != t[j]:  
                ok = False  
                break  
        if ok:  
            return i  
    return -1
```

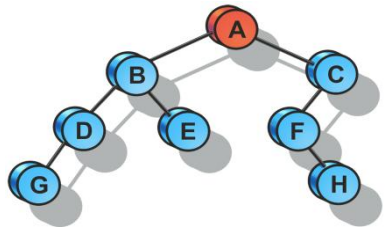


## Полиномиальная хеш-функция

Выберем константы  $P$  и  $x$

$$\text{hash}(S) = (s_0x^{n-1} + s_1x^{n-2} + \dots + s_{n-1}x + s_n) \bmod P$$

$s_i$  – код  $i$ -го символа строки  $S$



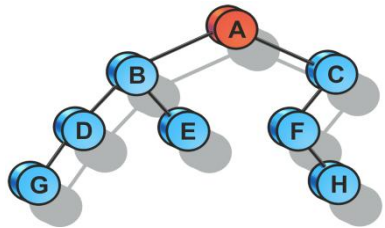
## Полиномиальная хеш-функция. Пример

Пусть  $x = 2$  и  $P = 5$

$$\text{hash}(\text{"hash"}) = (7 \cdot 8 + 0 \cdot 4 + 18 \cdot 2 + 7) \bmod 5 = 4$$

$$\begin{aligned} \text{hash}(\text{"string"}) &= (18 \cdot 32 + 19 \cdot 16 + 17 \cdot 8 + 8 \cdot 4 + \\ &+ 13 \cdot 2 + 6) \bmod 5 = 1080 \bmod 5 = 0 \end{aligned}$$

$$\begin{aligned} \text{hash}(\text{"computer"}) &= (2 \cdot 128 + 14 \cdot 64 + 12 \cdot 32 + \\ &+ 15 \cdot 16 + 20 \cdot 8 + 19 \cdot 4 + 4 \cdot 2 + 17) \bmod 5 = \\ &= 2037 \bmod 5 = 2 \end{aligned}$$



## Как вычислять хеш-функцию итеративно

$$\text{hash}(S) = (s_0x^{n-1} + s_1x^{n-2} + \dots + s_{n-1}x + s_n) \bmod P$$

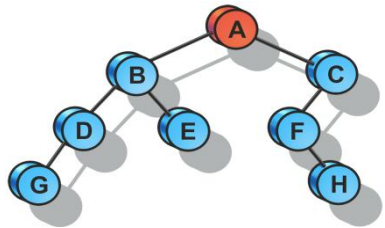
$$\begin{aligned}\text{hash}(S + c) &= (s_0x^n + s_1x^{n-1} + \dots + s_nx + c) \bmod P = \\ &= (\text{hash}(S) \cdot x + c) \bmod P\end{aligned}$$

$$\text{hash}(\text{"h"}) = 7$$

$$\text{hash}(\text{"ha"}) = (7 \cdot 2 + 0) \bmod 5 = 4$$

$$\text{hash}(\text{"has"}) = (4 \cdot 2 + 18) \bmod 5 = 1$$

$$\text{hash}(\text{"hash"}) = (1 \cdot 2 + 7) \bmod 5 = 4$$



## Быстрый пересчет хеш-функции для новой подстроки

$$H_i = \text{hash}(S[i : i + m])$$

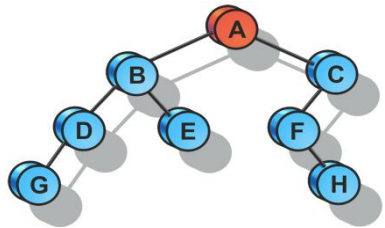
$$H_i = (s_i x^{m-1} + s_{i+1} x^{m-2} + \dots + s_{i+m-2} x + s_{i+m-1}) \bmod P$$

$$H_{i+1} = \text{hash}(S[i + 1 : i + 1 + m])$$

$$H_{i+1} = (s_{i+1} x^{m-1} + s_{i+2} x^{m-2} + \dots + s_{i+m-1} x + s_{i+m}) \bmod P$$

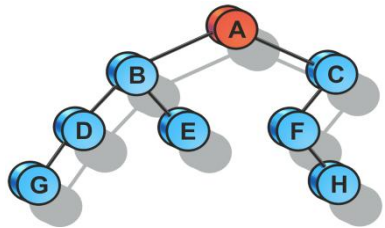
$$H_{i+1} = (H_i x - s_i x^m + s_{i+m}) \bmod P$$





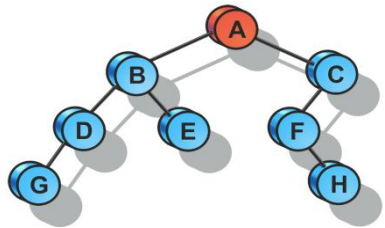
## Быстрый пересчет хеш-функции для новой подстроки. Код

```
buildH(s, t):  
    n = s.length  
    m = t.length  
    H[0] = hash(s[0 : m])  
    xx = powMod(x, m, P)  
    for i = 0 .. n - m - 1:  
        H[i + 1] = (H[i] * x - s[i] * xx  
                    + s[i + m]) % P
```



## Использование в поиске подстроки

```
find(s, t: String):  
    Ht = hash(t)  
    buildH(s, t)  
    for i = 0 .. s.length - t.length:  
        if H[i] != Ht:  
            continue  
        ok = True  
        for j = 0 .. t.length - 1:  
            if s[i + j] != t[j]:  
                ok = False  
                break  
        if ok:  
            return i  
    return -1
```



**Время работы**

$$O(N + M + NM/P)$$