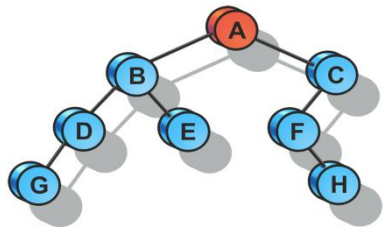


Алгоритмы программирования и структуры данных

Поиск подстрок

---

## Поиск подстрок (часть 4). Z-функция



## Z-функция

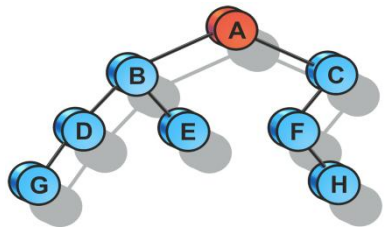
$i$

А В А В А В А С А В А

А В А В А В А С А В А

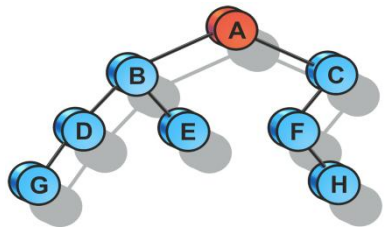
$z[i]$

$z[i]$  = длина наибольшей подстроки, начинающейся в позиции  $i$  и равной префиксу той же длины.



## Z-функция

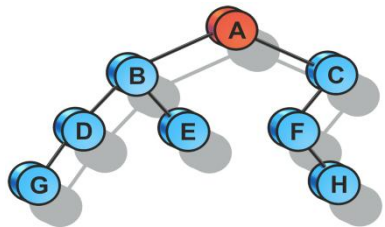
	A	B	A	B	A	B	A	C	A	B	A
Z	-	0	5	0	3	0	1	0	3	0	1



## Простой алгоритм

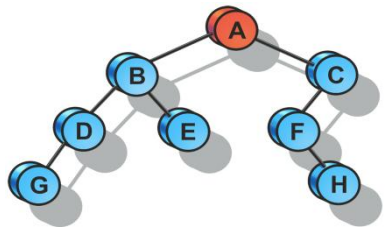
```
for i = 1..n - 1:  
    j = 0  
    while i+j < n && s[i+j] == s[j]:  
        j++  
    z[i] = j
```

Время работы:  $O(N^2)$



**Худший случай**

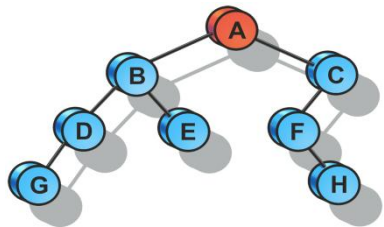
А А А А А А А А А А  
Z - 9 8 7 6 5 4 3 2 1



## Z-блоки

		┌──────────┐					┌──────────┐					
		└──────────┘					└──────────┘					
		А	В	А	В	А	В	А	С	А	В	А
Z	-	0	5	0	3	0	1	0	3	0	1	

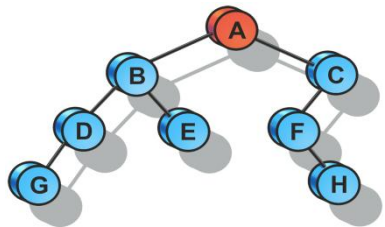
Z-блок – подстрока от  $i$  до  $(i + z[i] - 1)$ .



## Самый правый Z-блок

		L		R			
		└───┬───┘					
		А	В	А	В	А	В
Z	-	0	5	0	.	.	.

Храним z-блок  $[L..R - 1]$  с наибольшим R.



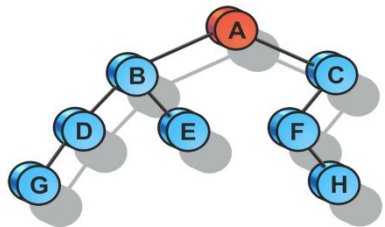
## Случай 1

$$i \geq R$$

				<sup>L</sup>	<sup>R</sup>						
				└─		<sup>i</sup>					
	A	B	A	C	A	B	A	C	A	B	A
Z	-	0	1	0	.	.	.	.	.	.	.

Жадно добавляем символы, как в простом алгоритме.



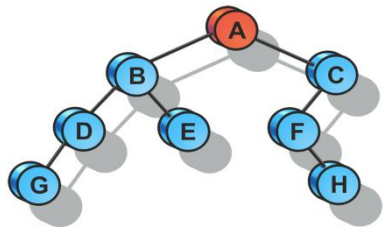


## Случай 1

$$i \geq R$$

					L	R			
					└──────────┘				
				i					
	A	B	A	C	A	B	A	C	A
Z	-	0	1	0	5	.	.	.	.

Обновляем L и R.

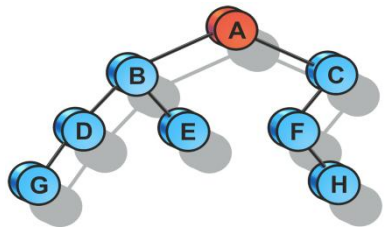


## Случай 2

$$i < R$$

$\overbrace{\hspace{10em}}^{L \hspace{1em} R}$   
 $\overset{i}{\text{АВАВАВАСАВА}}$   
 $Z \quad - \quad 0 \quad 5 \quad . \quad . \quad . \quad . \quad . \quad . \quad .$

Мы знаем про символы с номерами  $j = i..R-1$ , что  $S[j] = S[j - L]$ , их уже сравнивали в  $Z[i - L]$ .

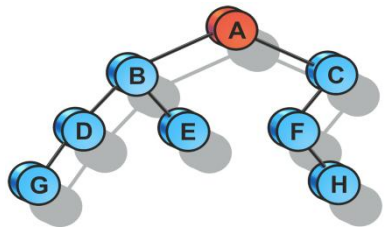


## Случай 2.1

$$z[i - L] < R - i$$

$\overbrace{\hspace{10em}}^{L \hspace{1em} R}$   
 $\overset{i}{\text{АВАВАВАСАВА}}$   
 $z \quad - \quad 0 \quad 5 \quad 0 \quad . \quad . \quad . \quad . \quad . \quad .$

$$z[i] = z[i - L].$$

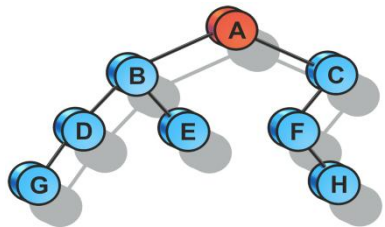


## Случай 2.2

$$z[i - L] \geq R - i$$

$\overbrace{\hspace{10em}}^{L \qquad \qquad \qquad R}$   
 $\overset{i}{\text{А В А В А В А С А В А}}$   
 $z \quad - \quad 0 \quad 5 \quad 0 \quad \textcolor{green}{3} \quad . \quad . \quad . \quad . \quad . \quad .$

$z[i]$  – не меньше  $R - i$ .

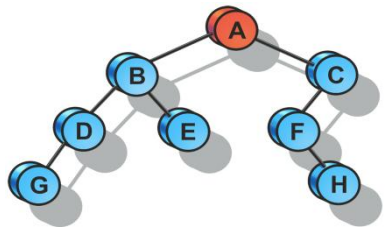


## Случай 2.2

$$z[i - L] \geq R - i$$

$\overbrace{\hspace{10em}}^{L \qquad R}$   
 $\overset{i}{\text{АВАВСАВАВАВ}} \text{АВС}$   
 $z \quad - \quad 0 \ 2 \ 0 \ 0 \ 4 \ 0 \ 5 \ . \ . \ . \ .$

Жадно добавляем символы, начиная с R.

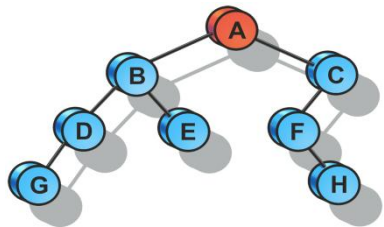


## Случай 2.2

$$z[i - L] \geq R - i$$

$L$   $R$   
 $i$   
 А В А В С А В А В А В С  
 z - 0 2 0 0 4 0 5 . . . .

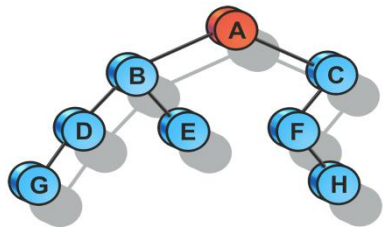
Обновляем  $L$  и  $R$ .



## Построение Z-функции. Код

```
L = 0, R = 0
for i = 1 .. n - 1
    if i >= R:
        j = 0
        while i + j < n && s[i + j] == s[j]:
            j++
        L = i, R = i + j
        z[i] = j
    else:
        if z[i - L] < R - i:
            z[i] = z[i - L]
        else:
            j = R - i
            while i + j < n && s[i + j] == s[j]:
                j++
            L = i, R = i + j
            z[i] = j
```

Время работы:  $O(N)$

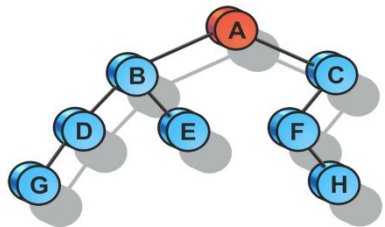


## Анализ времени работы

```
L = 0, R = 0
for i = 1 .. n - 1
    if i >= R:
        j = 0          // R <= i + j
        while i + j < n && s[i + j] == s[j]:
            j++        // R увеличивается
        L = i, R = i + j
        z[i] = j
    else:
        if z[i - L] < R - i:
            z[i] = z[i - L]
        else:
            j = R - i    // R = i + j
            while i + j < n && s[i + j] == s[j]:
                j++      // R увеличивается
            L = i, R = i + j
            z[i] = j
```

Время работы:  $O(N)$





## Поиск подстроки

В А В А В С А В А С В А D  
 А В А С

А В А С # В А В А В С А В А С В А D  
 Z - 0 1 0 0 0 3 0 2 0 0 4 . . . . .