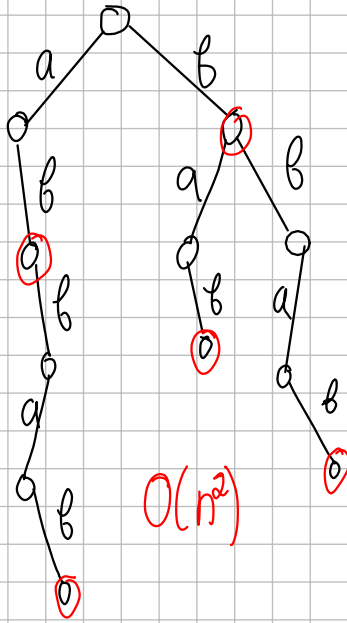


abba

Строим бор,  
который  
содержит  
все суф-сы

То есть теперь можно  
за линию проверять,  
есть ли слово в тексте

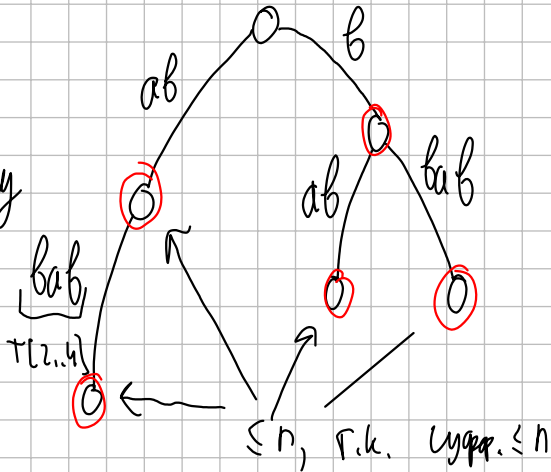
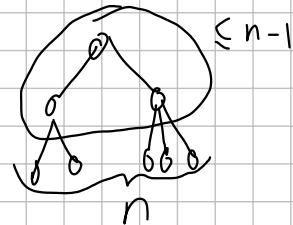
Каждая вершина бора - какая-то подстрока текста, т.к. это префикс суффикса

 $O(n^2)$ 

Сожмем дерево. Терминальные пока сжимать не будем

0 1 2 3 4  
a b b a b

хранит пару  
индексов \

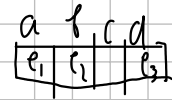
 $O(n)$ 

Node:

Edge[] children // или мапа Символ->Ребро  
Node suf link

Edge:

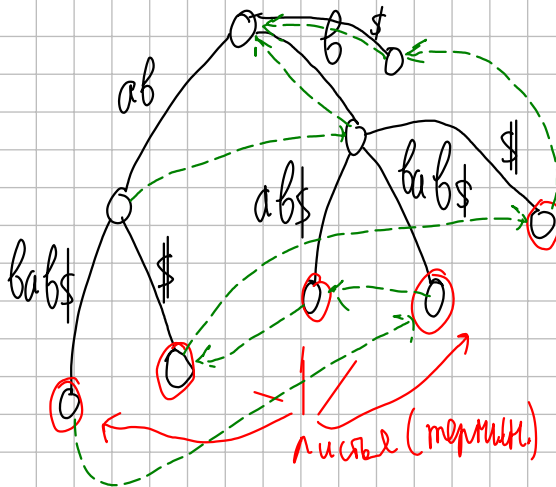
```
int L, R
Node dest
```



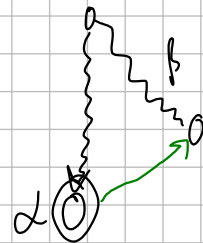
Наше дерево уже линейного размера.  
Осталось его быстро построить.  
Алгоритм называется "Алгоритм Укконена"

$$\overline{T} = abbab \$$$

$\alpha$ -ногемона

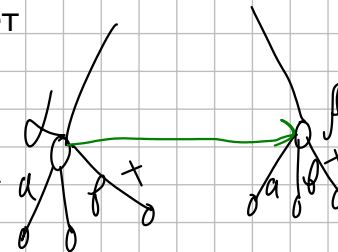


суф. сепка



$\beta$ -суп. 2

 $|\beta| \rightarrow \max$ 
$$|\beta| = |\alpha| - 1$$



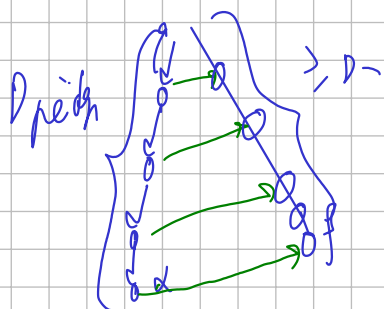
type 2:  
edge, pos

```
if (type == 1)
    type = 2
    edge = node.children(c)
    pos = 1
else
    pos++
if pos == edge.len
    type = 1
    node = edge.dest
```

 $O(1)$ 

A hand-drawn diagram on a grid background. At the top left, a node is labeled  $\alpha$ . A line goes down from  $\alpha$  to another node at the bottom left. A horizontal line goes from  $\alpha$  to the right, ending at a node labeled  $\beta$ . Below  $\alpha$ , there is a horizontal line segment with a red arrow pointing up to it, and a Greek letter  $\gamma$  is written above it. To the right of  $\beta$ , there is a vertical line with a node in the middle, and a diagonal line going down from  $\beta$  to another node. Various blue and red annotations are present, including arrows and Greek letters like  $\gamma$ ,  $\beta$ , and  $\alpha$ .

точно есть. Поэтому мы  
сможем спуститься вниз,  
не переходя еще раз по суф  
ссылке



$D = \text{depth}(\text{ur\_state})$   
(4 nodes)

$n$ 

$$\begin{array}{l} \text{cur\_state} = \text{root} \\ \text{go(a)} \\ \text{go(b)} \\ \text{suf\_link()} \\ \text{go()} \\ \text{go()} \\ \text{suf\_link()} \\ \text{suf\_link()} \\ \dots \end{array}$$

$$\begin{array}{l} \Delta D \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ \geq -1 + k_1 \\ \geq 0 \\ \geq 0 \\ \geq -1 + k_2 \\ \geq -1 + k_2 \end{array}$$

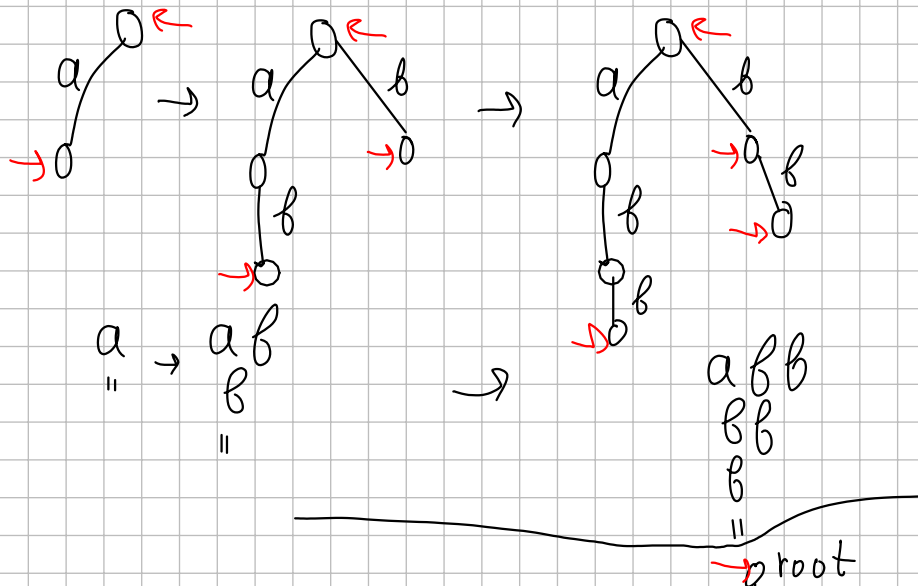
$$n \geq D \geq -n + \sum k_i$$

$$\sum T = O(n)$$

$$\boxed{\sum k_i \leq 2n}$$

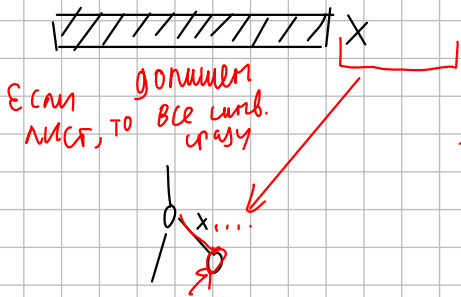
Строим бор. По одному символу добавляем

a b b a b a b b \$



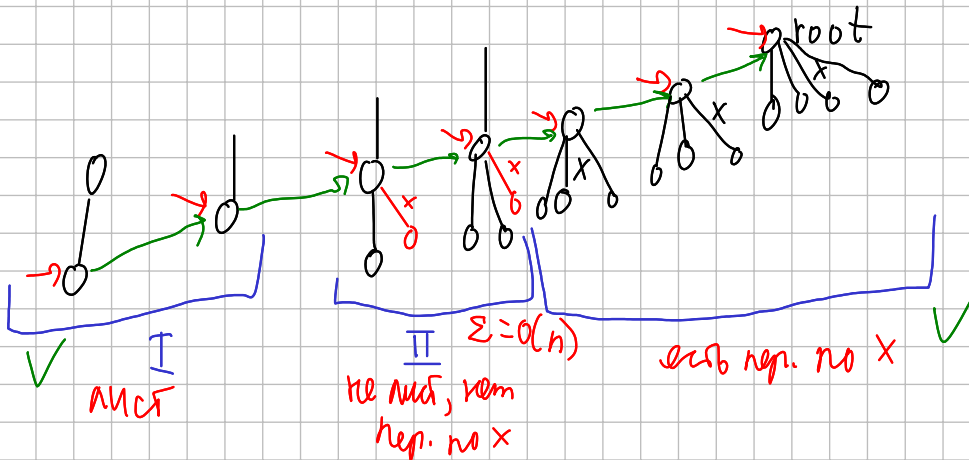
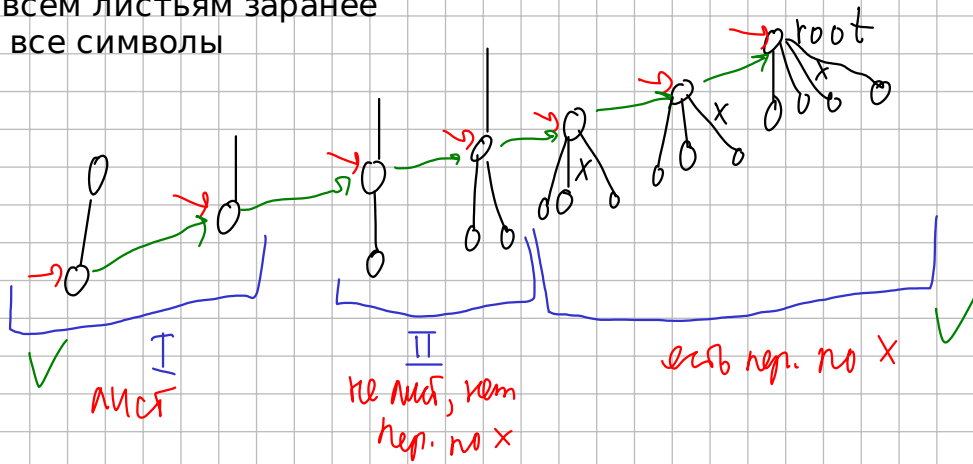
Так делать долго,  
надо быстрее

a b b a b a b b \$



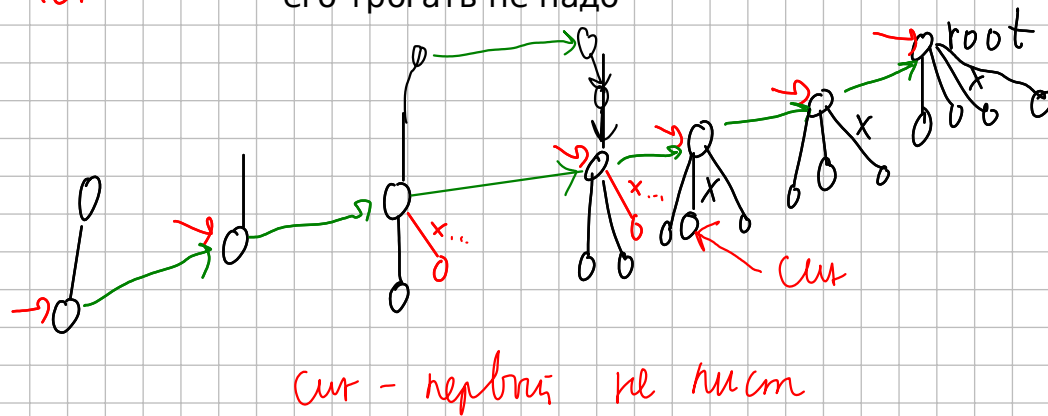
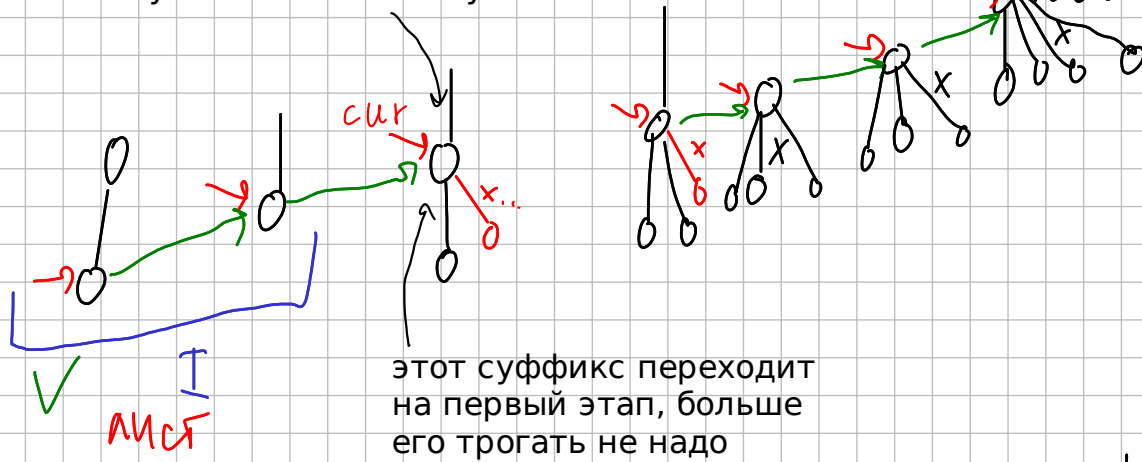
Если есть переход по x, то у всех следующих он тоже есть (т.к. они наши суффиксы)

То есть ко всем листьям заранее припишем все символы



Дерево в конце  $O(n)$  размера

возможно указывает на сжатую



Есть указатель на cur - первый суффикс второго этапа.

Все предыдущие уже листья

Добавляем 'x':

Берем текущую вершину. Если нет перехода по 'x', отпочковываем, получили лист, переносим его на первый этап, переходим к следующей. Повторяем, ...

Нашли первую вершину, у которой уже есть переход по 'x'. Переходим по 'x', и здесь останавливаемся. Новый этап мы начнем отсюда.

На каждом действии мы переходим либо вниз по 'x', либо по суффиксной ссылке. Все линейно, суммарно значит линейно.

Мы постепенно разжимаем вершины-суффиксы, и от них протягиваем суффиксные ссылки.

Пример

abababab\$

