

Print All Palindromic Substrings (23 July)

Str = "L I L I L"

str.substr(⁰1, ²j)

[i, j)

str = "L I L I L"
 0 1 2 3 4 5 6 7 8 9 10 11
 ↓ ↓
 [1, 4]

str.substr(2, 6) = "LILIL"
 ↓ ↓ ↓ ↓
 (2, 8) (5, 11)

str = "012"
 (Lib . Coding)

str.substr(i)

i → n-1

i = 2

Lib Coding

0 1 2 3 4

Str - n i t i n

len = 5

$$\text{Substrings} = \frac{n(n+1)}{2} = \frac{5(5+1)}{2} = 15$$

0
 n
 n i
 n i t
 n i t i
 n i t i n

2
 t
 t i
 t i n

3
 i
 i n

4
 n

Str--

0	1	2	3	4
n	9	7	9	n
	i			
	j			

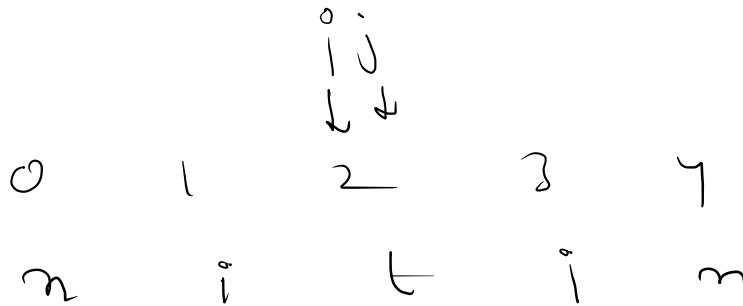
String str = str.substr(i, j)
(0, 1)

if (is Palind (str)) \rightarrow return

i = 0 to n-1

j = i+1 to n

is Palindrom



$i = 0$
 $j = n - 1$

$ch_i == ch_j \rightarrow i++ \quad j--$
 $ch_i \neq ch_j \rightarrow \text{not palindrom}$

↓

Palindromic

$\rightarrow i < j$

str =

o i l j
n i t a n

ch_i

ch_j

i ≠ a → not palindrome
↳ break from loop

Method 2 → using group → w/o using sorting str

Ex- a b a b a c b c d

? ↓ SortX → time comp

a a a a b b b c c

a 1 c
a 4 b 3 c 2

String Builder

- mutable
- non Primitive
- more eff in terms of
space

Syntax

⇒ `StringBuilder sb = new StringBuilder ()`

`StringBuilder sb = new StringBuilder (" I like Java")`

`StringBuilder sb = new StringBuilder();`

↳ empty string

Stacks → a data structure

↓
ab
string

