



strings °

distinct Palindromic Subsequence °

{ str = "ab,c p cba"

alpha, numeric char...  
a-z or 0-9

lexographical

a, abba, b, baab, aa, c

sort

str = abbaab

✓ a  
ab  
abb  
✓ abba  
abbaab  
abbaa

✓ b  
✓ bb  
bba  
bbaa  
bbaab

✓ b  
ba  
baa  
✓ baab

✓ a  
✓ aa  
aab

✓ a  
ab

✓ a  
✓ abba  
✓ b  
bb  
x b  
✓ baab

x a  
✓ aa  
x a  
x b

str = "abba"

Subseq:  $\checkmark a$   $\checkmark b$   $\checkmark b$   $\checkmark a$   
ab bb  
abba bba

all the palindromic subseq.:  $\{\checkmark a, \checkmark abba, \checkmark b, \checkmark bb, \checkmark a, \checkmark b\}$

lexographical order:  $\{\checkmark a, a, \checkmark abba, \checkmark b, b, \checkmark bb\}$   
(collection.sort())

↓ filter all distinct palindromic subseq.

ans  $\{a, abba, b, bb\}$

steps

- ✓ ① generate all subseq.
  - ✓ ② store all palindromic subseq in a AL.
  - ✓ ③ lexicographically sort all palindromic subseq.
  - ✓ ④ filter all unique/distinct subseq.
-

# Strings

↳ split()

String

splits the given string over the string passed

str = "ab|cd|ef|gh"

String[] arr = str.split(",");

✓ arr = { "ab", "cd", "ef", "gh" }

str = "A B C B D F M O J D B M"

A diagram illustrating the splitting of the string "A B C B D F M O J D B M" by the character 'B'. Vertical lines separate the segments: "A", "B", "C", "B", "D F M O J D", "B", and "M". Yellow arrows above the lines indicate the direction of the split: right for the first line, left for the second, right for the third, left for the fourth, right for the fifth, left for the sixth, and right for the seventh. Yellow checkmarks are placed below the lines for the segments "A", "C", "D F M O J D", and "M", indicating they are the resulting substrings after removing the 'B' characters.

String[] arr = str.split("B");

arr = { "A", "C", "D F M O J D", "M" }

str = "A M O A B C N A B C P G R"

A diagram illustrating the splitting of the string "A M O ABC N ABC P G R" by the substring "ABC". Vertical lines separate the segments: "A", "M", "O", "ABC", "N", "ABC", "P", "G", and "R". Yellow arrows above the lines indicate the direction of the split: right for the first line, left for the second, right for the third, left for the fourth, right for the fifth, left for the sixth, and right for the seventh. The substrings "ABC" and "ABC" are underlined in blue. This indicates that the resulting array will contain the segments "A M O", "N", and "P G R" after removing the "ABC" substrings.

String[] arr = str.split("ABC");

arr = { "A M O", "N", "P G R" }

# String value to Integer

Str = "0123"  
1234

\_\_\_\_\_

→ Convert to int value.

9874

Put a = ~~0~~;  $0 \times 10 + 9 = 9$

for (int i = 0; i < str.length(); i++)

{ char ch = str.charAt(i);

✓ int value = (int)(ch - '0');

a = a \* 10 + value;

}

ascii '0' → x  
'1' → x+1  
'2' → x+2  
⋮  
So on

String to Integer

→ String str = "1234"

Put a = Integer.parseInt(str);

✓ a = 1234  
→



str = "4897";  
int a = 0;

✓ for (int i = 0; i < str.length(); i++)  
{  
 char ch = str.charAt(i);  
 int value = (int)(ch - '0');  
 a = a \* 10 + value;  
}

ch = '4'

v = 4

$$a = 0 * 10 + 4 = 4$$

ch = '8'

v = 8

$$a = 4 * 10 + 8 = 48$$

ch = '9'

v = 9

$$a = 48 * 10 + 9$$

= 489



# Time Conversion

Imp Base  
★  
12hr Based  
12:00:00 PM  
24hr Based  
12:00:00

Imp Base  
★  
12hr Based  
12:00:00 AM  
24hr Based  
00:00:00

12hr Based  
05:24:36 PM  
24hr Based  
17:24:36

12hr Based  
07:24:56 AM  
24hr Based  
07:24:56

★ if we have PM  
adding 12 Pn hr

String in 12 hr Based Timing

→ Convert to 24 hr Based Timing!

05:56:24 PM

str = hh:mm:ss PM

↑↑

↑

int { hh = 5;

5 + 12 = 17

Ptice

0 1 2 3 4 5 6 7 8  
str = "A A A A B B B B B"

Adrian's -  
A B C A B C A B C  
0 1 2 3 4 5 6 7 8

A  $\rightarrow$  0, 3, 6  $\rightarrow i \% 3 = 0$

B  $\rightarrow$  1, 4, 7  $\rightarrow i \% 3 = 1$

C  $\rightarrow$  2, 5, 8  $\rightarrow i \% 3 = 2$

→ ✓ 0 1 2 3 4 5 6 7 8

str = "A A A A B B B B B"  
↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

$0 \% 3 = 0$  → 'A'

$1 \% 3 = 1$  → 'B'

$2 \% 3 = 2$  → 'C'

$3 \% 3 = 0$  → 'A'

$4 \% 3 = 1$  → 'B'

$5 \% 3 = 2$  → 'C'

$6 \% 3 = 0$  → 'A'

$7 \% 3 = 1$  → 'B'

$8 \% 3 = 2$  → 'C'