

String

Sequence of char value

char 1 Byte

Name = "Priya";

String

0	1	2	3	4	5
P	r	i	y	a	\0

terminating character in a string

Reversal of String

$s = \{ 'h', 'e', 'l', 'l', 'o' \}$

Reverse string $s = \{ 'o', 'l', 'l', 'e', 'h' \}$

Don't we

built-in

functions in this situation

reverse();

Two Pointers Approach

① Reverse in a
Given String

$s = \{ "h", "e", "l", "l", "o" \}$

left = 0 right = 4

Time complexity
 $O(n)$

{ while (left < right)
 swap (s(left), s(right));
 left++;
 right--
 }
 return s;

Space complexity $\rightarrow O(1)$

$Str = "Priya-Bhatia"$

0	1	2	3	4	5	6	7	8	9	10	11	12
P	r	i	y	a	-	B	h	a	t	i	a	-

$Str.length()$



Removes the extra

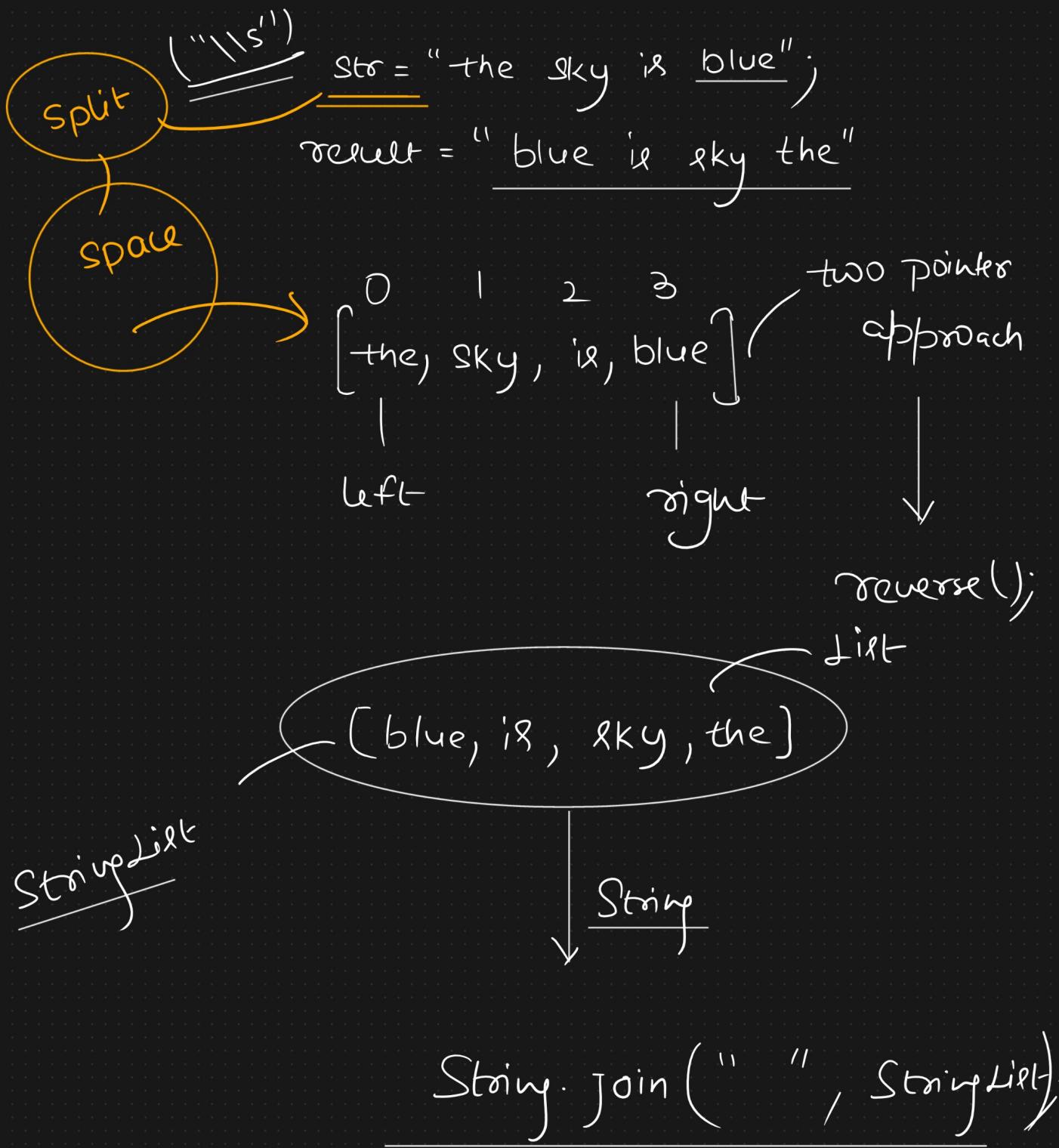
whitespace

$$\left\{ \begin{array}{l} Str1 = 7 \\ Str2 = 5 \end{array} \right\} \rightarrow 7 + 5 = 12$$

$$7 + 5 - 1$$
$$m + n - 1$$

A_{str1}
7

A_{str2}
5



String = "the — — sky — — is — blue"

list

Space in the

story

[the, sky, is, blue]

Maximum Subarray



(i, j)
 Permutation
 k combinations

maximum
Subarray

max. =

$O(n^3)$

$\text{for } (i = 0 \text{ to } n-1) \text{ do }$

$\text{for } (j = i+1 \text{ to } n-1) \text{ do }$

Sum + = $G(j)$

$\text{for } (k \rightarrow (i, j))$

Sum + =

$\max(\underline{\text{max}},$
Sum)

4

return max;

16

max = 2

)

sum = 2 8 5

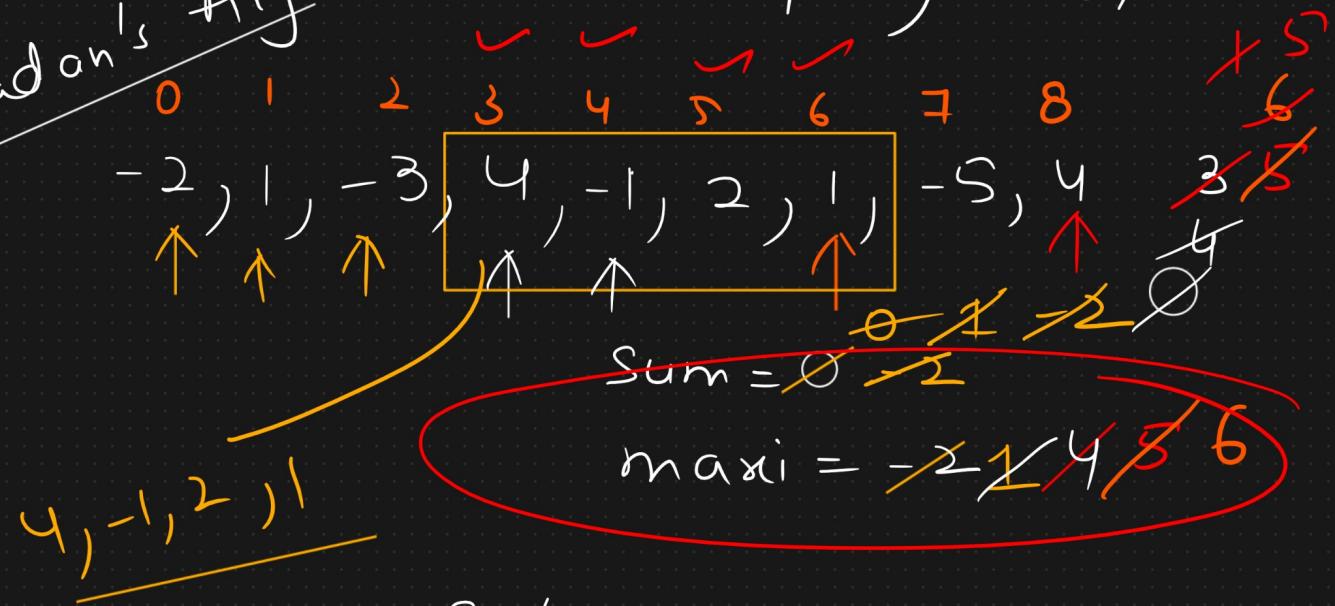
8 16

16

[2, 4, -1, 3, 8]

Dynamic Programming

Kadane's Algorithm time complexity = $O(n)$



for ($i = 0$ to $n-1$)

Sum += arr(i);

if (Sum >= maxi)

maxi = Sum;

Negative if (Sum < 0)

Sum = 0;

return maxi

Y

DSA → Optimization Problem



Maximum & Minimum



Greedy Algorithms

Dynamic Programming

traverse the same

Path / subarray again

Longest
Substring
Without
Repeating
Characters

Str = abc abc b b

3

HashMap / Map

Python

Dictionary { }

<key, value> Pair

EmpId, Salary

Value

unique key

Unique key

{ Set = remove all the
 duplicate values
 in the given
 data structure