

5) → Valid parenthesis -

1. Question -

Given a string $(, [, {,], },)$ determine if the string is valid (or) not.

Valid if -

- Open brackets must be closed by the same type of brackets $() \quad \{ \}$ $[]$
- Open brackets must be closed in the correct order.

$[(())] \rightarrow [()] \rightarrow \text{" "}$ $(] \times$
 $[()] \rightarrow \text{" "}$
 $[] \times$

2. Examples -

$S = ()$ ✓
→ true

$S = () [] \{ \}$ " "
→ true

$S = (]$
→ false

$\{ [[] \{ \}] \} ([])$
Valid subexpression

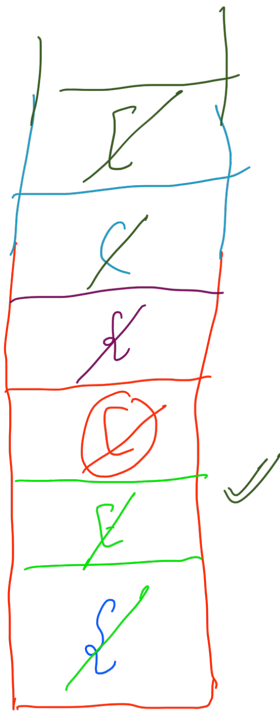
3. Intuition - / Solution / →

$(\{ \{ [([]) \{ \}] \} ([]))$ " "
Valid subexpression

- will be one big valid expression
- (or) will be multiple valid sub-expressions

This holds recursive structure \dots

$s = \{ \{ [[(]) \} \} \}$



$((((())) \rightarrow \text{false}$

Map

A diagram illustrating a deformation of a loop. On the left, a blue loop is shown with a red arrow pointing to a red loop on the right. The red loop is a simplified version of the blue loop, with the blue loop's structure being more complex and the red loop being more compact.

return time

Time = $O(n)$ $\rightarrow n$

Space = $O(n)$ / Stack size $\rightarrow n$?

CCd CCd CCd...

$$\begin{array}{r} 1 \\ 1 \\ \hline 2 \\ \hline C \\ \hline C \end{array}$$

what if there is only one type of bracket?

Handwritten text: CC CC CC CC CC CC

$T = O(n)$
 $S = O(1)$

CC ?
D

count Op P ↓

$$\boxed{\text{count} + Q_p P \uparrow} \mid \approx 0 \text{ \&\& } \boxed{\text{C}}$$

6. Code walk through -

boolean isValid(S) {
 Stack<Character> stack...

for (Character bracket : S) {

if (map.containsKey(bracket)) {

 Character top = stack.isEmpty() ?

 'X' : stack.peek();

if (top != map.get(bracket)) {

return false;

 }

 stack.pop();

 } else {

 stack.push(bracket);

 }

}

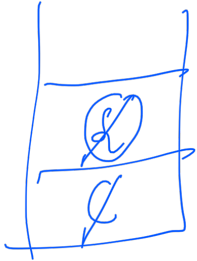
return stack.isEmpty();

true : false

map

{	}
[]
{	}
[]

{ { } }



}