# Longest Valid Parenthesis :-

```
         0  1  2  3  4  5
S :    E ( ( ) ) ( )
         •  •  •  •

maxlen = 2 4 6

if  s[i] == '(' :
        stack.append(i)
else :
        stack.pop()  x
        if  len(s) == 0 :
                    x x

        else :
            maxlen = max (maxlen, i - stack[-1])
```

-1

```
⏚ 1  2  3 4  5 ⑥ 7  8                    8
) ( ) ( ) ( )  (  (
↑ ↑ ↑ ↑ ↑ ↑↑ ↑↑   ↑                      7
                                         8
⤷  [ ) ) ( ) ( ( ) ) ( ) ) (              1
        ⊙                                 0
if  s[i] == 'C' :
    stack.append(i)

else:                          ┌──────────────────┐
    stack.pop() ✓              │  max = 2 4 6      │
    if  len(s) == 0 :•         └──────────────────┘
        stack.append(i)                    ↗
                               0 1 2 3 4 5
                               ( ) ( ) ( )
                               ↑        ↑
                                   ⑤
w ( — else: P)
    maxlen = max (maxlen,  i - stack[-1])

 5 - 0 + 1
```

| 2 | 1 | 5 | 6 | 2 | 3 |
|---|---|---|---|---|---|



nser

nsel
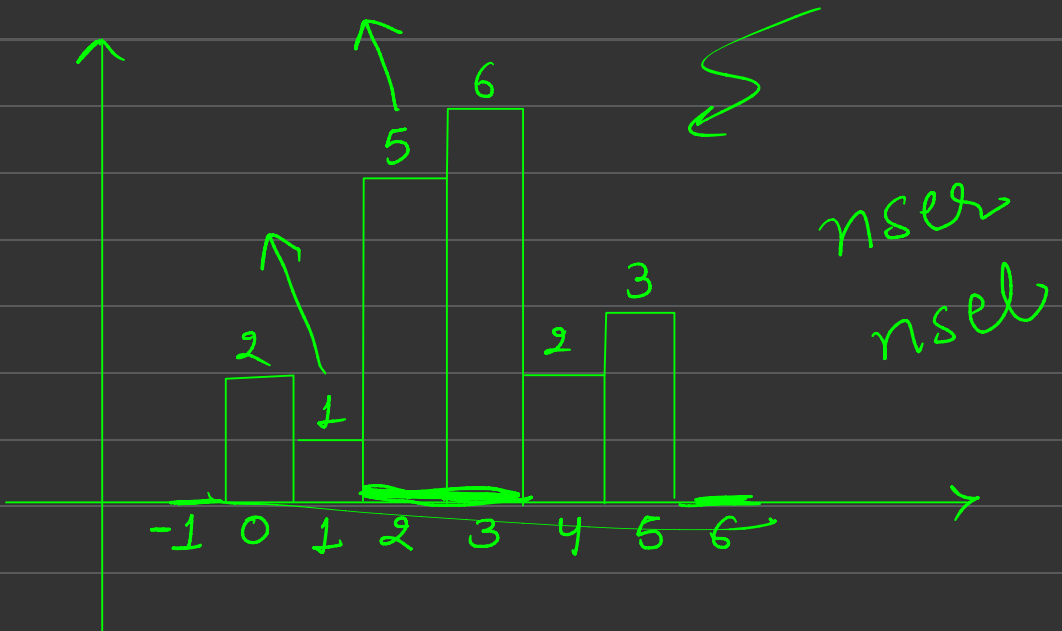
$\textcircled{5}$ : left : 1 , right : 4

$$4 - 1 - 1 = \textcircled{2} = 10$$

$\textcircled{1}$ . left : -1 , right : 6

$$6 - (-1) - 1 \Rightarrow 6 = 6$$

|  | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|--|----|---|---|---|---|---|---|---|
|  |    | ②| ①| ⑥| 6 | 2 | 3 | ↑ |

nsr

| 1 | 6 | 4 | 4 | 6 | 6 |
|---|---|---|---|---|---|

nsl

| -1 | -1 | 1 | 2 | 1 | 4 |
|----|----|---|---|---|---|

①    ⑥    ②

2    6    10

nsr — nsl — 1

[ ] (nsr) (heights)

| [ | | | ] |
|---|---|---|---|

[ ] (nsl) (heights)