

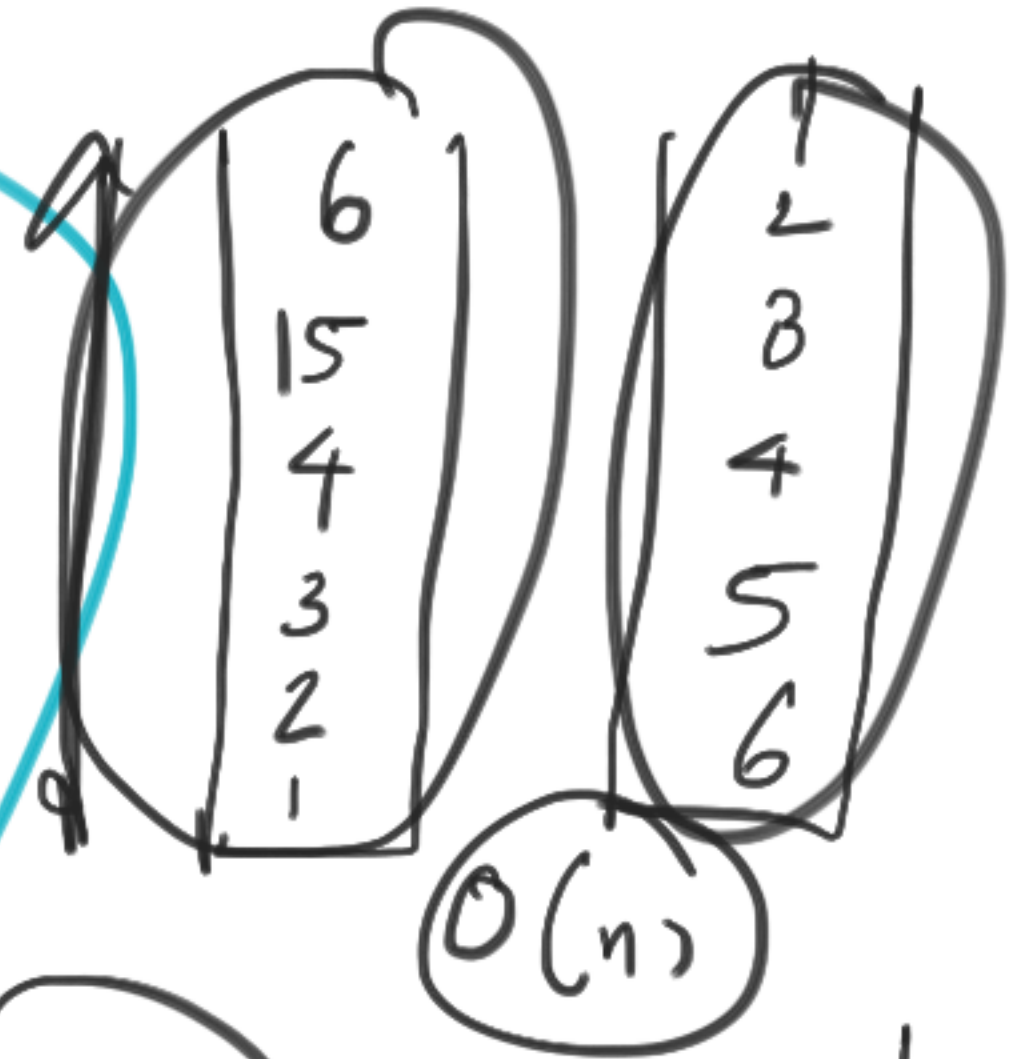
while (count > 0)

{ temp.push(s.top())

s.pop()

count = count - 1

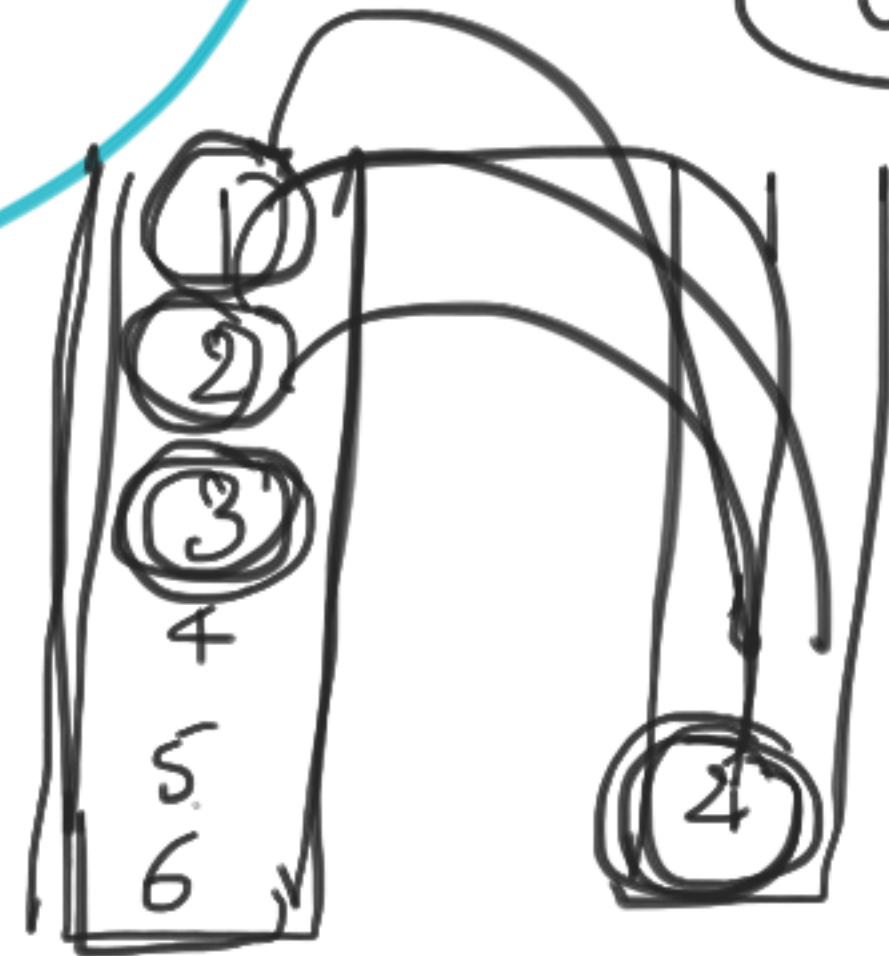
}



Worst Case Scenario

$1 + 2 + 3 + \dots + n$

$O(n^2)$



1
2
3
4
5
6

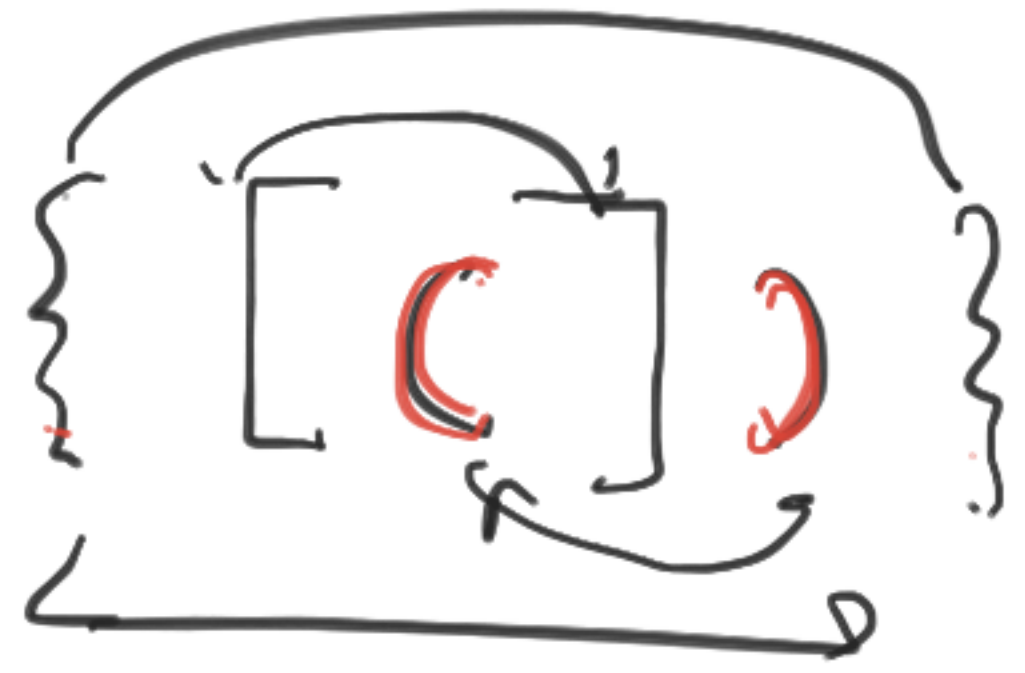
Sort stack using recursion. (?)

Valid parentheses

expression

balanced expression

() { } []



X

order

()

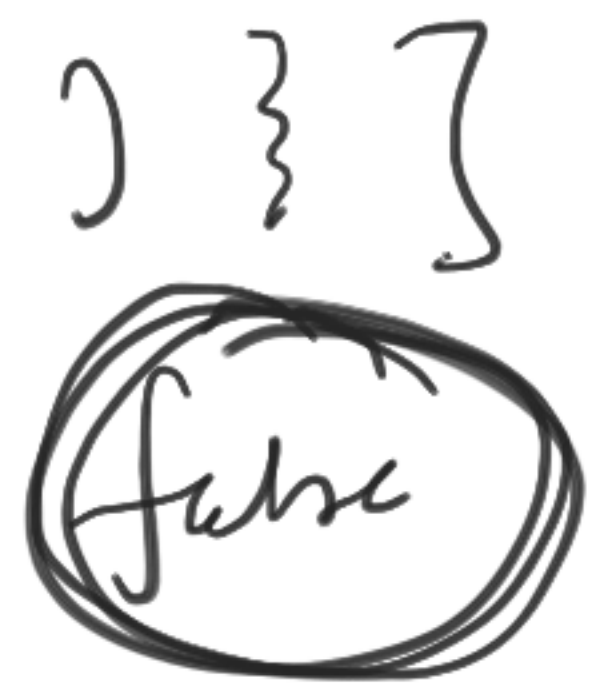
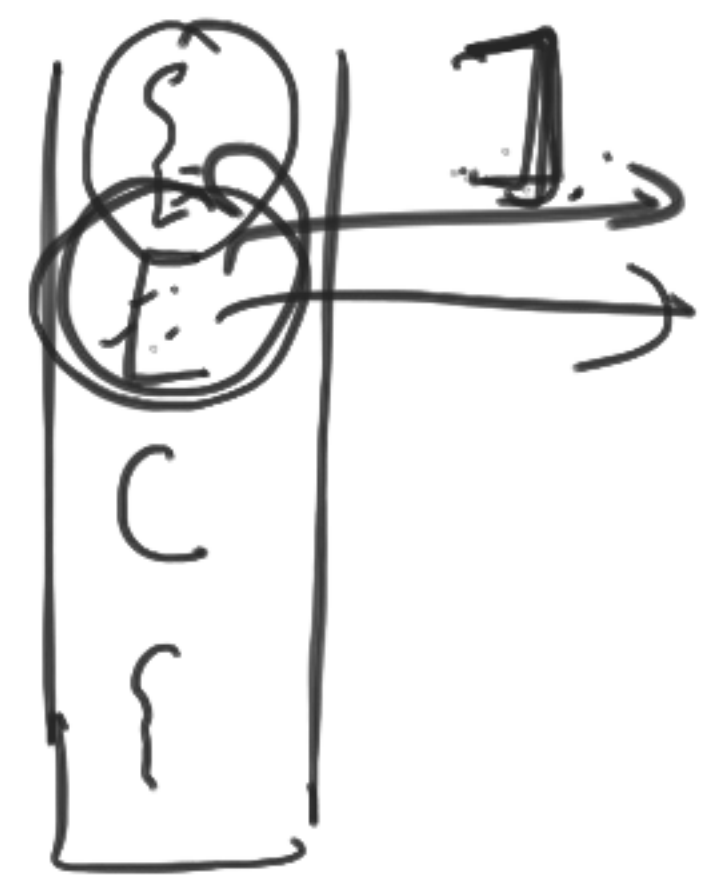
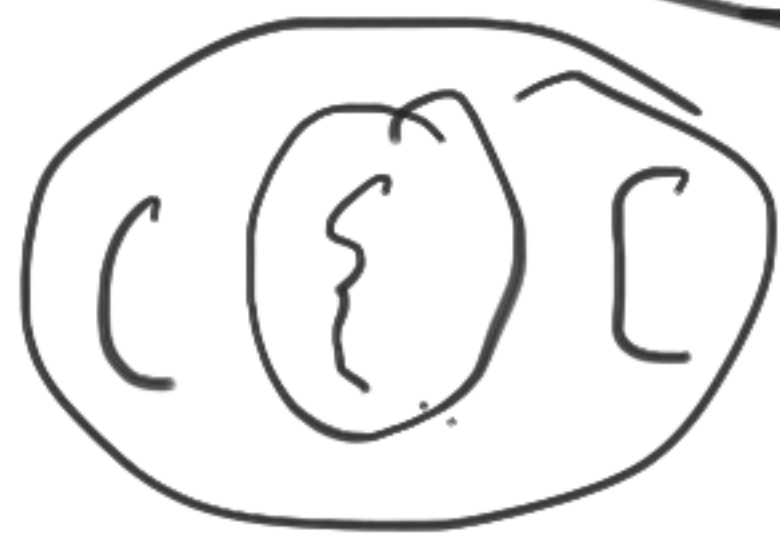
LIFO

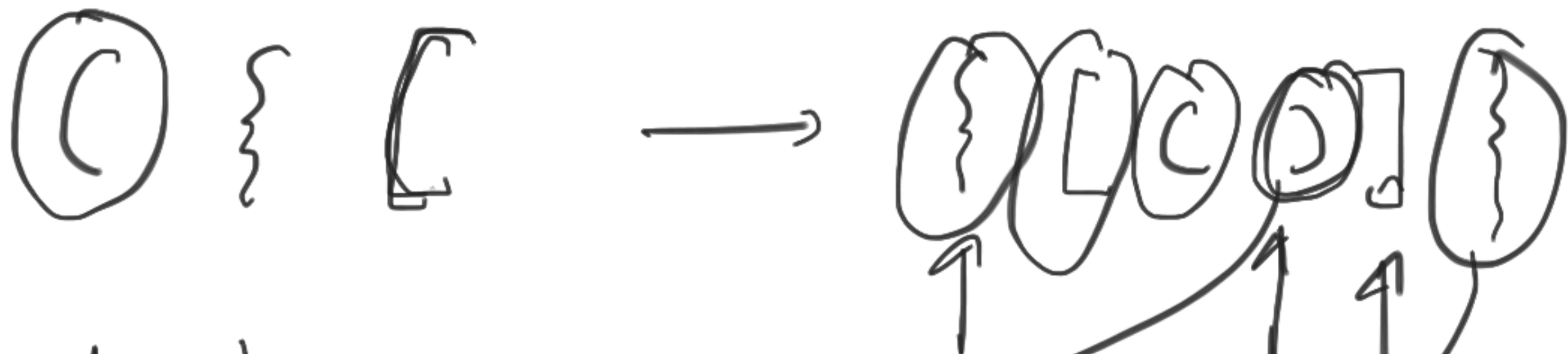
{ } [] { }

exp []
n
 $n \cdot 1 \cdot 2 \cdot 1 = 0$
odd



stack
must
be empty





for ($i = 0$ to $exp.size() - 1$)



{
if ($exp[i] == C$)

$s.push()$

$= = \{$

$s.push()$

$= = \{$

$s.push()$

else if

$!s.empty()$

$\Delta \Delta (s.top() == exp[i])$

$s.pop()$

else

$\{ return false \}$

$\{ return (s.empty()) \}$

((()) () ())

without
using stack
①

((())) (

unbalanced

⇒ min. no. of removal

min no. of additions

balanced



() ())) (()) (())

✓ longest valid parentheses.



$\max(x, y, z)$

Generate all balanced parentheses.

of size n

$$\Rightarrow 60$$

③

A hand-drawn diagram consisting of a large circle. Inside the circle, at the top, are two 'n' characters. At the bottom, there are two curved lines, one on the left and one on the right, resembling a smiley face.

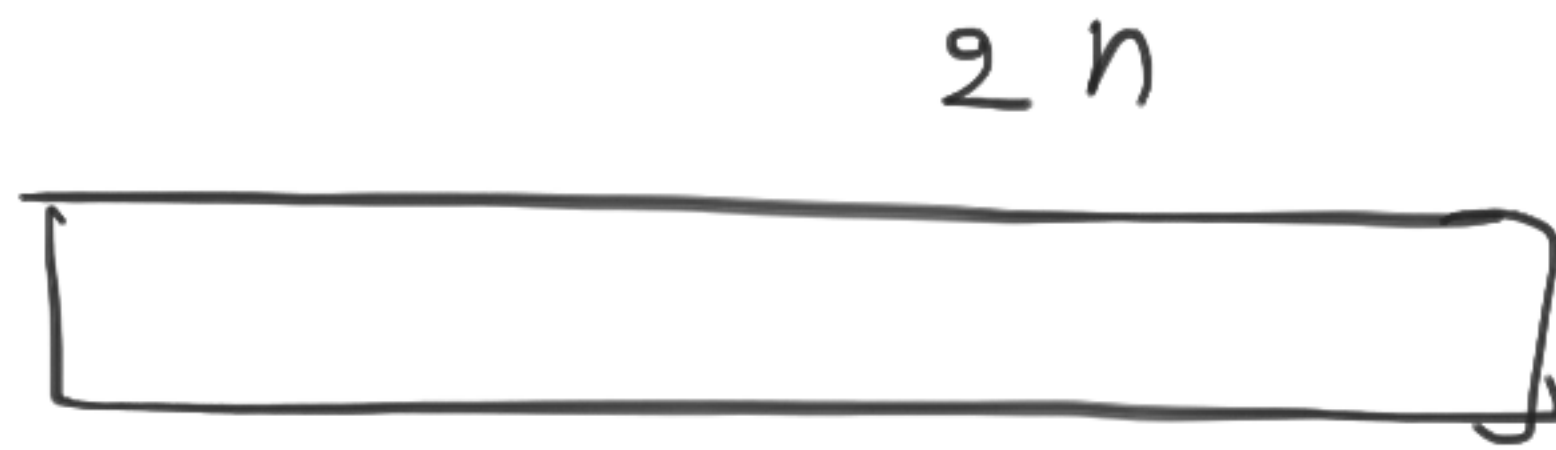
$$\left\{ \begin{array}{l} (()) \\ () () \end{array} \right.$$
$$((C, C))$$

(,) (\ ()

~~(C)~~

(()) ()

() (())



min swap

expression
balanced

