**STACK TEMPLATES**

**✔ Template 1: Monotonic Stack**

stack = []

for i in reversed(range(len(arr))): # or forward

while stack and stack[-1] <= arr[i]:

stack.pop()

res[i] = stack[-1] if stack else -1

stack.append(arr[i])

🧠 Used in:

* Next Greater Element I/II
* Daily Temperatures
* Stock Span

**✔ Template 2: Classic Push/Pop Matching**

stack = []

for ch in s:

if ch in "([{":

stack.append(ch)

else:

if not stack or not is\_match(stack[-1], ch):

return False

stack.pop()

return not stack

🧠 Used in:

* Valid Parentheses
* Expression validation

**✔ Template 3: Stack of Tuples**

stack = []

for num in nums:

curr\_min = min(num, stack[-1][1]) if stack else num

stack.append((num, curr\_min))

🧠 Used in:

* Min Stack (LeetCode 155)
* Max Stack

**✔ Template 4: Index-Based Stack**

stack = []

for i in range(len(heights)):

while stack and heights[stack[-1]] > heights[i]:

height = heights[stack.pop()]

width = i if not stack else i - stack[-1] - 1

max\_area = max(max\_area, height \* width)

stack.append(i)

🧠 Used in:

* Largest Rectangle in Histogram (LeetCode 84)
* Maximal Rectangle (LeetCode 85)

**✔ Template 5: Undo Simulation**

stack = []

for ch in s:

if stack and stack[-1] == ch:

stack.pop()

else:

stack.append(ch)

return "".join(stack)

🧠 Used in:

* Remove Adjacent Duplicates
* Text editor simulations