Stacks-2

Assignment Solution

PW SKILLS

1. Valid Parentheses [Leetcode - 20]

Solution

```
class Solution:
    def isValid(self, s: str) → bool:
        stack = []
        mapping = {")":"(", "}":"{", "]":"["}

        for char in s:
            if char in mapping.values():
                stack.append(char)
            elif char in mapping.keys():
                if not stack or mapping[char] ≠ stack.pop():
                    return False

        return not stack
```

2. Next Greater Node in Linked List [Leetcode - 1019]

Solution

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def reverseList(self, head: Optional[ListNode]) →
Tuple[Optional[ListNode], int]:
        pre = None
        l = 0
        while head:
            l += 1
            cur = head
            head = cur.next
            cur.next = pre
            pre = cur

        return pre, l

    def nextLargerNodes(self, head: Optional[ListNode]) →
List[int]:
        new_head, l = self.reverseList(head)
        res = [0] * l

 stack  = []
        while new_head:
            l -= 1
            while stack and stack[-1] ⩽ new_head.val:
                stack.pop()

            if stack:
                res[l] = stack[-1]

            stack.append(new_head.val)
            new_head = new_head.next

        return res
```

3. Final Prices with a Special Discount in a Shop [Leetcode - 1475]

Solution

```
from typing import List

class Solution:
    def finalPrices(self, prices: List[int]) -> List[int]:
        n = len(prices)
        res = [0] * n
        for i in range(n):
            flag = False
            for j in range(i + 1, n):
                if prices[j] <= prices[i]:
                    res[i] = prices[i] - prices[j]
                    flag = True
                    break
            if not flag:
                res[i] = prices[i]

        return res
```

4. Next Greater Element II [Leetcode - 503]

Solution

```
from collections import deque
class Solution:
    def nextGreaterElements(self, nums: List[int]) -> List[int]:
        length = len(nums)
        nums.extend(nums)
        stack = deque()
        ans = []
   print(nums)
        for i in range(len(nums)-1,-1,-1):
            while len(stack)>0 and stack[len(stack)-1]<=nums[i]:
                stack.pop()
            if len(stack)==0:
                ans.append(-1)
            else:
                ans.append(stack[len(stack)-1])
            stack.append(nums[i])
        ans = ans[::-1]
        return ans[:length]
```