

## 1. Two Sum

class Solution:

```
def twoSum(self, nums: List[int], target: int) -> List[int]:
    memo = {}
    for i, num in enumerate(nums):
        dif = target - num
        if dif in memo:
            return [memo[dif], i]
        else:
            memo[num] = i
```

## 53. Maximum Subarray

class Solution:

```
def maxSubArray(self, nums: List[int]) -> int:
    result = 0
    max_res = -float('inf')
    for num in nums:
        result = max(result + num, num)
        max_res = max(result, max_res)
    return max_res
```

## 21. Merge Two Sorted Lists

class Solution:

```
def mergeTwoLists(self, l1: ListNode, l2: ListNode) -> ListNode:
    dummy_head = ListNode(0)
    dummy_head_memo = dummy_head
    while l1 and l2:
        if l1.val < l2.val:
            dummy_head.next = l1
            l1 = l1.next
        else:
            dummy_head.next = l2
            l2 = l2.next
        dummy_head = dummy_head.next

    if l1:
        dummy_head.next = l1
    elif l2:
        dummy_head.next = l2

    return dummy_head_memo.next
```

## 20. Valid Parentheses

```

class Solution(object):
    def isValid(self, s):
        """
        :type s: str
        :rtype: bool
        """
        table = {'(': ')', '{': '}', '[': ']'}
        length = len(s)
        stack = []
        for i in range(length):
            if s[i] in '({[':
                stack.append(s[i])
            else:
                if not stack or table[stack.pop()] != s[i]:
                    return False
        return len(stack) == 0

```

### 937. Reorder Data in Log Files

```

class Solution:
    def reorderLogFiles(self, logs: List[str]) -> List[str]:
        letters = []
        digits = []
        digits_map = '1234567890'
        for log in logs:
            if log.split(' ')[1][0] in digits_map:
                digits.append(log)
            else:
                letters.append(log)
        letters.sort(key=lambda a: (a[a.index(' ') + 1:], a))
        return letters + digits

```