PROBLEM:1

Problem: Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

Solution (Python):

python

def two_sum(nums, target):

num_map = {}

for i, num in enumerate(nums):

complement = target - num

if complement in num_map:

return [num_map[complement], i]

num_map[num] = i

return []

Example

2. Reverse a Linked List

print(two_sum([2, 7, 11, 15], 9)) # Output: [0, 1]

Problem: Given the head of a singly linked list, reverse the list and return the new head. **Solution (Python):**

```
python

class ListNode:

def __init__(self, val=0, next=None):

self.val = val

self.next = next

def reverse_list(head):

prev, curr = None, head

while curr:

next_node = curr.next

curr.next = prev

prev = curr

curr = next_node

return prev
```

3. Valid Parentheses

```
Problem: Given a string s containing just the characters ()[]{}, determine if the input string is valid.
Solution (Python):
python
def is_valid(s):
  stack = []
  mapping = {')': '(', '}': '{', ']': '['}
  for char in s:
    if char in mapping:
       top = stack.pop() if stack else '#'
       if mapping[char] != top:
         return False
    else:
       stack.append(char)
  return not stack
# Example
print(is_valid("()[]{}")) # Output: True
print(is_valid("(]")) # Output: False
4. Find the Missing Number
Problem: Given an array nums containing n distinct numbers in the range [0, n], return the missing number.
Solution (Python):
```

```
python

def missing_number(nums):
    n = len(nums)
    return n * (n + 1) // 2 - sum(nums)

# Example
print(missing_number([3, 0, 1])) # Output: 2
```

5. Merge Two Sorted Lists

Problem: Merge two sorted linked lists and return the sorted list. **Solution (Python):** python

```
class ListNode:
  def __init__(self, val=0, next=None):
    self.val = val
    self.next = next
def merge_lists(l1, l2):
  dummy = ListNode()
  current = dummy
  while I1 and I2:
    if l1.val < l2.val:
      current.next = I1
      11 = 11.next
    else:
      current.next = I2
      12 = 12.next
    current = current.next
  current.next = I1 if I1 else I2
  return dummy.next
6. Longest Substring Without Repeating Characters
Problem: Given a string s, find the length of the longest substring without repeating characters.
Solution (Python):
python
def length_of_longest_substring(s):
  char_map = {}
  left = max_length = 0
  for right, char in enumerate(s):
    if char in char_map:
      left = max(left, char_map[char] + 1)
    char_map[char] = right
    max_length = max(max_length, right - left + 1)
  return max_length
```

Example

print(length_of_longest_substring("abcabcbb")) # Output: 3

7. Find First and Last Position of Element in Sorted Array

Problem: Given a sorted array of integers and a target value, find the first and last position of the target. Solution (Python): python def search_range(nums, target): def binary_search(left): I, r = 0, len(nums) - 1while I <= r: mid = (I + r) // 2if nums[mid] > target or (left and nums[mid] == target): r = mid - 1else: I = mid + 1return I left = binary_search(True) return [left, binary_search(False) - 1] if left <= binary_search(False) - 1 else [-1, -1] # Example print(search_range([5, 7, 7, 8, 8, 10], 8)) # Output: [3, 4] 8. Rotate an Array

```
Problem: Given an array, rotate the array to the right by k steps.
Solution (Python):

python

def rotate(nums, k):
    k %= len(nums)
    nums[:] = nums[-k:] + nums[:-k]

# Example

arr = [1, 2, 3, 4, 5, 6, 7]

rotate(arr, 3)

print(arr) # Output: [5, 6, 7, 1, 2, 3, 4]
```

9. Check if a Number is Palindrome

```
Problem: Given an integer x, return True if x is a palindrome.

Solution (Python):

python

def is_palindrome(x):

return str(x) == str(x)[::-1]

# Example

print(is_palindrome(121)) # Output: True

print(is_palindrome(-121)) # Output: False
```

10. Find the Intersection of Two Arrays

```
Problem: Given two integer arrays, return their intersection (unique values). Solution (Python):
```

```
python

def intersection(nums1, nums2):
   return list(set(nums1) & set(nums2))

# Example
```

print(intersection([1, 2, 2, 1], [2, 2])) # Output: [2]