1. def get\_indices(nums: list[int], target: int) -> list[int]:

temp = {}

for i, x in enumerate(nums):

if x in temp:

return [i, temp[x]]

temp[target - x] = i

2. def removeElement(nums: list[int], val: int) -> int:

collect = []

for i in nums:

if i == val:

collect.append(i)

for i in collect:

nums.remove(i)

return len(nums)

3. def searchInsert(nums: list[int], target: int) -> int:

l = 0

r = len(nums)

while l <= r:

mid = l + (r-l)//2

if nums[mid] == target:

return mid

elif nums[mid] < target:

l = mid + 1

elif nums[mid] > target:

r = mid - 1

return l

4. def plusOne(digits: list[int]) -> list[int]:

n = len(digits)

num = 0

for i in range(n):

power = n - i - 1

num += digits[i]\*10\*\*power

num += 1

return [int(x) for x in str(num)]

5. def merge(nums1: list[int], m: int, nums2: list[int], n: int) -> None:

"""

Do not return anything, modify nums1 in-place instead.

"""

for i in range(len(nums1) - m):

nums1.remove(0)

nums1.extend(nums2)

nums1.sort()

return

6. def containsDuplicate(nums: list[int]) -> bool:

nums\_set = set()

for n in nums:

if n in nums\_set:

return True

nums\_set.add(n)

return False

7. def moveZeroes(nums: list[int]) -> None:

"""

Do not return anything, modify nums in-place instead.

"""

n = len(nums)

c = 0

for i in range(n):

if nums[i] != 0:

nums[c] = nums[i]

c += 1

for i in range(c, n):

nums[i]=0

return

8. def findErrorNums(nums: list[int]) -> list[int]:

n = len(nums)

sum\_of\_nums\_actual = n\*(n+1)//2

sum\_of\_nums\_given = sum(nums)

nums\_set = set()

for x in nums:

if x in nums\_set:

return [x, sum\_of\_nums\_actual-sum\_of\_nums\_given+x] \

if sum\_of\_nums\_given < sum\_of\_nums\_actual \

else [x, x-(sum\_of\_nums\_given-sum\_of\_nums\_actual)]

nums\_set.add(x)