**ICS 2105 – Data Structures and Algorithms**

**NJENGA BEATRICE WAMBUI**

**SCT212-0153/2022**

**QUIZ 1**

**Question One (1) Remove Duplicates from Sorted Array**

Given a sorted array nums, remove the duplicates in-place such that each element appears only once and returns the new length.

Do not allocate extra space for another array, you must do this by modifying the input array in-place with O(1) extra memory.

def remove\_duplicates(nums):

if not nums:

return 0

# Initialize pointers

i = 1 # Pointer for iteration

j = 1 # Pointer for updating unique elements

# Iterate through the array

while i < len(nums):

# If current element is different from previous one

if nums[i] != nums[i - 1]:

# Update the array with the unique element

nums[j] = nums[i]

# Move the updating pointer forward

j += 1

# Move the iteration pointer forward

i += 1

return j # Length of the array with unique elements

# Example usage:

nums = [1, 1, 2, 2, 3, 4, 4, 4, 5]

new\_length = remove\_duplicates(nums)

print(new\_length) # Output: 5

print(nums[:new\_length]) # Output: [1, 2, 3, 4, 5]

**Question Two (2) - Rotate Array**

Given an array, rotate the array to the right by *k* steps, where *k* is non-negative.

def rotate\_array(nums, k):

# Handle cases where k is greater than the length of the array

k %= len(nums)

# Reverse the entire array

nums.reverse()

# Reverse the first k elements

nums[:k] = reversed(nums[:k])

# Reverse the remaining elements

nums[k:] = reversed(nums[k:])

# Example usage:

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

k = 3

rotate\_array(nums, k)

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

**Question Three (3) - Contains Duplicate**

Given an array of integers, find if the array contains any duplicates.

Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

def contains\_duplicate(nums):

seen = set()

for num in nums:

if num in seen:

return True

seen.add(num)

return False

# Example usage:

nums\_with\_duplicates = [1, 2, 3, 4, 5, 1]

nums\_without\_duplicates = [1, 2, 3, 4, 5]

print(contains\_duplicate(nums\_with\_duplicates)) # Output: True

print(contains\_duplicate(nums\_without\_duplicates)) # Output: False

**Question Four (4) - Single Number**

Given a non-empty array of integers nums, every element appears twice except for one. Find that single one.

def single\_number(nums):

result = 0

for num in nums:

result ^= num

return result

# Example usage:

nums = [2, 1, 4, 4, 5, 3, 2, 1, 3]

print(single\_number(nums)) # Output: 5