

# CSE 2216 : Data Structures and Algorithm 1 Lab

## Problem Set for Searching

### 1: Linear Search

**Problem:** Given an array of integers, find the first occurrence of a target integer. If the target is not in the array, return -1.

**Example:**

**Input:** arr = [5, 3, 8, 1, 9], target = 8

**Output:** 2

### 2: Linear Search with Multiple Occurrences

**Problem:** Given an array, find all the indices where a target integer appears. Return an array of indices. If the target does not appear, return an empty array.

**Example:**

**Input:** arr = [4, 2, 3, 2, 4, 2], target = 2

**Output:** [1, 3, 5]

### 3: Linear Search with Condition

**Problem:** Find the first element in an array of integers that is greater than a given target. If no such element exists, return -1.

**Example:**

**Input:** arr = [3, 5, 7, 2, 8, 10], target = 6

**Output:** 7

### 4: Binary Search in a Sorted Array

**Problem:** Implement binary search in a sorted array to locate a target value. Return the index of the target if found; otherwise, return -1.

**Example:**

**Input:** arr = [1, 3, 5, 7, 9], target = 5

**Output:** 2

## 5: Binary Search in Descending Order Array

**Problem:** Perform binary search on a descending order sorted array to find a target value.

**Example:**

**Input:** arr = [9, 7, 5, 3, 1], target = 7

**Output:** 1

## 6: Binary Search for First and Last Occurrence

**Problem:** Given a sorted array, find the first and last positions of a target value. If the target is not found, return (-1, -1).

**Example:**

**Input:** arr = [1, 2, 2, 2, 3, 4], target = 2

**Output:** (1, 3)

## 7: Find the Insert Position

**Problem:** Given a sorted array, return the index where a target value should be inserted to maintain the order. Use binary search.

**Example:**

**Input:** arr = [1, 3, 5, 6], target = 4

**Output:** 2

## 8: Count Occurrences of Target with Binary Search

**Problem:** In a sorted array, count the occurrences of a target value using binary search.

**Example:**

**Input:** arr = [2, 4, 4, 4, 6, 7], target = 4

**Output:** 3

## 9: Find Closest Element with Binary Search

**Problem:** Given a sorted array, find the element closest to a given target. If two elements are equally close, return the smaller one.

**Example:**

**Input:** arr = [1, 3, 8, 10, 15], target = 12

**Output:** 10

## 10: Find Peak Element with Binary Search

**Problem:** Given an array where elements increase and then decrease (a "mountain" array), find the index of the peak element using binary search.

**Example:**

**Input:** arr = [1, 3, 8, 12, 4, 2]

**Output:** 3 (peak element 12 is at index 3)