# Binary Search

LeetCode offers a variety of problems that involve **binary search** techniques, which are commonly used to solve problems efficiently when the input is sorted or when the problem can be reduced to a range search. Below is a list of some common **binary search** problems on LeetCode:

## 1. Binary Search

- **Problem**: Given a sorted array of integers, return the index of the target value if it exists, otherwise return -1.
- **Difficulty**: Easy
- **Example**:

```
Input: nums = [-1,0,3,5,9,12], target = 9
Output: 4

```

- **Solution**: Implement a binary search algorithm.

## 2. Search Insert Position

- **Problem**: Given a sorted array and a target value, return the index where the target is located. If the target is not present, return the index where it should be inserted in order to keep the array sorted.
- **Difficulty**: Easy
- **Example**:

```
Input: nums = [1,3,5,6], target = 5
Output: 2

```

## 3. Find Minimum in Rotated Sorted Array

- **Problem**: Suppose an array of length `n` sorted in ascending order is rotated between `1` and `n` times. Find the minimum element of the array.
- **Difficulty**: Medium
- **Example**:

```
Input: nums = [3,4,5,1,2]
Output: 1

```

## 4. Find Peak Element

- **Problem**: A peak element in an array is an element that is strictly greater than its neighbors. Find any peak element and return its index.
- **Difficulty**: Medium
- **Example**:

```
Input: nums = [1,2,3,1]
Output: 2

```

### 5. First Bad Version

- **Problem**: You are given an API `isBadVersion(version)` which returns a boolean indicating whether the version is bad or not. You need to find the first bad version.
- **Difficulty**: Easy
- **Example**:

```
Input: n = 5, bad = 4
Output: 4

```

### 6. Search in Rotated Sorted Array

- **Problem**: Given a rotated sorted array and a target value, search for the target in the array and return its index. If not found, return -1.
- **Difficulty**: Medium
- **Example**:

```
Input: nums = [4,5,6,7,0,1,2], target = 0
Output: 4

```

### 7. Search in Rotated Sorted Array II

- **Problem**: This is similar to the above problem, but this time, the array may contain duplicates. Search for the target in the array and return its index.
- **Difficulty**: Medium
- **Example**:

```
Input: nums = [2,5,6,0,0,1,2], target = 0
Output: 3

```

### 8. Kth Smallest Element in a Sorted Matrix

- **Problem**: You are given an `n x n` matrix where each of the rows and columns is sorted in ascending order. Find the kth smallest element in the matrix.
- **Difficulty**: Medium
- **Example**:

```
Input: matrix = [[1,5,9],[10,11,13],[12,13,15]], k = 8
Output: 13

```

### 9. Median of Two Sorted Arrays

- **Problem**: Given two sorted arrays, find the median of the two sorted arrays.
- **Difficulty**: Hard
- **Example**:

```
Input: nums1 = [1, 3], nums2 = [2]
Output: 2.0

```

## 10. Capacity To Ship Packages Within D Days

- **Problem**: You are given a list of weights and need to ship all packages within `D` days. Find the minimum weight capacity of the ship required to carry all the packages within `D` days.
- **Difficulty**: Medium
- **Example**:

```
Input: weights = [1,2,3,4,5,6,7,8,9,10], D = 5
Output: 15

```

## 11. Pow(x, n)

- **Problem**: Implement `pow(x, n)` which calculates `x` raised to the power `n` (i.e., `x^n`).
- **Difficulty**: Medium
- **Example**:

```
Input: x = 2.00000, n = 10
Output: 1024.00000

```

## 12. Binary Search Tree Iterator

- **Problem**: Implement an iterator over a binary search tree (BST) that traverses the tree in order.
- **Difficulty**: Medium
- **Example**:

```
Input:
["BSTIterator", "next", "next", "hasNext", "next", "hasNext"]
[[[7,3,15,null,null,9,20]], [], [], [], [], []]
Output: [null, 3, 7, true, 9, true]

```

## 13. Median of a Data Stream

- **Problem**: Design a data structure that supports the following operations:
  - `insertNum(int num)`: Insert a number into the data stream.
  - `findMedian()`: Return the median of all elements inserted so far.
- **Difficulty**: Hard
- **Example**:

```
Input: ["MedianFinder", "insertNum", "insertNum", "findMedian", "insertNum", "findMedian"]
Output: [null, null, null, 1.0, null, 2.0]

```

---

## How Binary Search is Used:

In many of these problems, binary search is applied in the following ways:

- Searching in a **sorted array** (finding an element or an index).
- **Finding the minimum/maximum** in a rotated or modified sorted array.
- **Dividing the search space** (e.g., when finding the kth smallest element, or the optimal capacity to ship packages).

- **Optimization problems** where the search space is continuous or discrete (e.g., finding the smallest value that satisfies a certain condition).

These problems are perfect for practicing binary search and learning how to apply it in different contexts.