Smart Coding & Interview Series
Top-20 Basic Program
(Binary Search Thinking)

First, understand the solution building strategies and coding for the problems in LIVE/VIDEO session and then you apply those strategies discussed in LIVE/VIDEO session to solve the following problems. Use your favourite language(C/C++/Java/C#/Python/Scala) for coding.

1) Guess Number: We are playing the Guess Game. The game is as follows: I pick a number from 1 to n. You have to guess which number I picked. Every time you guess wrong, I'll tell you whether the number is higher or lower. You call a pre-defined API guess(int num) which returns 3 possible results (-1, 1, or 0):

-1: My number is lower

1: My number is higher

0: Congrats! You got it!

Example :

Input: n = 10, pick = 6

Output: 6

Source : *https://leetcode.com/problems/guess-number-higher-or-lower/description/*

2) Ceil: Given a sorted array and a target value, return the index if the target is found. If not, return the index of the next immediate higher element in the array. You may assume no duplicates in the array.

Example :

Input: [1,3,5,6], 4

Output: 2

3) Search in a Sorted Array of Unknown Size: Given an integer array sorted in ascending order, write a function to searchtargetinnums. Iftargetexists, then return its index, otherwise return-1.However, the array size is unknown to you. You may only access the array using anArrayReader interface, where ArrayReader.get(k)returns the element of the array at indexk (0-indexed). You may assume all integers in the array are less than 10000, and if you access the array out of bounds,ArrayReader.getwill return2147483647.

Example:

Input:

array = [-1,0,3,5,9,12], target = 9

Output: 4

Source: *https://leetcode.com/problems/search-in-a-sorted-array-of-unknown-size/description/*

Smart Coding & Interview Series
Top-20 Basic Program
(Binary Search Thinking)

4) Matrix Search-II: Write an efficient algorithm that searches for a value in an *m* x *n* matrix. This matrix has the following properties:

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

Example:

Consider the following matrix:

```
[
  [01,04,07,11,15],
  [02,05,08,12,19],
  [03,06,09,16,22],
  [10,13,14,17,24],
  [18,21,23,26,30]
]
```

Given target = 5, return true.

Given target = 20, return false.

Source: *https://leetcode.com/problems/search-a-2d-matrix-ii/description/*

5) Rotated Array Search-I(No Duplicates): Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. (i.e., [0,1,2,4,5,6,7] might become [4,5,6,7,0,1,2]). You are given a target value to search. If found in the array return its index, otherwise return -1. You may assume no duplicate exists in the array.

Example:

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