1.题型范式：

1.1一个有序的数据结构（例如数组） 1.2搜索问题 1.3要求以logn或者近似logn的时间复杂度解决问题

2.解题模板：

这类题目，最重要的就是首先划分好题目有的几种情况，例如一些具体的例子来试验，从而确定以下的几个步骤

2.1确定左右边界

要使得搜索的左右边界的闭区间，left和right，一定要锁住答案

2.2确定退出规则

要使得退出时，左右边界范围已经缩小到锁定某个元素

2.3确定边界缩小规则

关键在于，在应用二分的思想，即通过middle这个点，每次以类似减少一半的搜索范围，在达到log的时间复杂度的同时，每次缩小左右边界仍然可以锁住答案

2.4锁定答案位置

在搜索某个具体的target时，只要按照题目要求，返回找到target此时的位置索引或bool值即可。在更复杂的情况，没有要求搜索某个的target，而是搜索某种模式，例如类升序数组中的“山谷”，通常做法是通过划分好题目有的几种情况，通过具体的例子来试验，最后来确定退出搜索时，答案位于left或者right还是其他什么位置。

3.常见变式：

3.1.数组并不是严格的升降序，甚至有可能只有极松散的规律。此时，仍然要优先考虑二叉搜索。

3.2.边界缩小时，不对称，甚至不能够减小一半的搜索范围，只能某些情况下减小例如1一个索引位置的搜索范围，退化为线性。

3.3并不是在一维数组中搜索，而是在矩阵甚至是其他数据结构中搜索，此时一种直接的方法就是运用转换的思想，将一维数组和给定的数据结构进行转换。

例题1：

[704. 二分查找 - 力扣（Leetcode）](https://leetcode.cn/problems/binary-search/submissions/372935668/)

例题2：

[153. 寻找旋转排序数组中的最小值 - 力扣（LeetCode）](https://leetcode.cn/problems/find-minimum-in-rotated-sorted-array/submissions/)

例题3：

[162. 寻找峰值 - 力扣（Leetcode）](https://leetcode.cn/problems/find-peak-element/description/)  
例题4：

[74. 搜索二维矩阵 - 力扣（Leetcode）](https://leetcode.cn/problems/search-a-2d-matrix/description/)

例题5：

[240. 搜索二维矩阵 II - 力扣（Leetcode）](https://leetcode.cn/problems/search-a-2d-matrix-ii/description/)