

二分查找 Binary Search

在一个已经排好序的数组中查找某个目标值

每次比较目标值和当前区间的中间元素，逐步缩小一半的搜索区间

时间复杂度是 $O(\log n)$

操作：

假设我们在一个升序数组 $A[1, \dots, n]$ 中查找目标值 x :

1] 设定两个指针 $low = 1, high = n$

2] 计算中点: $mid = \lfloor (low + high) / 2 \rfloor$

3] 比较 $A[mid]$ 与 x :

- if $A[mid] == x \rightarrow \text{find}, \text{return } mid$
- if $A[mid] > x \rightarrow \text{目标在左边}, \text{let } high \leftarrow mid - 1$
- if $A[mid] < x \rightarrow \text{目标在右边}, \text{let } low \leftarrow mid + 1$