二分查找（Binary Search）

一、算法简介

二分查找（Binary Search），是一种在有序数组中查找某一特定元素的查找算法。查找过程从数组的中间元素开始，如果中间元素正好是要查找的元素，则查找过程结束；如果某一特定元素大于或者小于中间元素，则在数组大于或小于中间元素的那一半中查找，而且跟开始一样从中间元素开始比较。如果在某一步骤数组为空，则代表找不到。

这种查找算法每一次比较都使查找范围缩小一半。

二、思路分析

三、复杂度分析

时间复杂度：折半搜索每次把搜索区域减少一半，时间复杂度为 O(logn)

空间复杂度：O(1)

注：折半查找的前提条件是需要有序表顺序存储，对于静态查找表，一次排序后不再变化，折半查找能得到不错的效率。但对于需要频繁执行插入或删除操作的数据集来说，维护有序的排序会带来不小的工作量，那就不建议使用。

说明：元素必须是有序的，如果是无序的则要先进行排序操作

技巧

分析二分查找的一个技巧是：不要出现 else，而是把所有情况用 else if 写清楚，这样可以清楚地展现所有细节。

代码中 left + (right - left) / 2 就和 (left + right) / 2 的结果相同，但是有效防止了 left 和 right 太大直接相加导致溢出。