

思路:

1. 确定目标元素和有序数组。
2. 初始化搜索范围为整个数组。
3. 比较目标元素与中间元素。
 - 如果相等, 返回中间元素的索引, 查找成功。
 - 如果目标元素小于中间元素, 缩小搜索范围为左半部分。
 - 如果目标元素大于中间元素, 缩小搜索范围为右半部分。
4. 重复步骤2和步骤3, 直到找到目标元素或搜索范围为空。查找成功返回目标元素的索引, 否则返回-1。

这个算法的时间复杂度为 $O(\log n)$, 适用于有序数组的快速查找操作。

代码实现:

```
1  class Solution {
2      public int search(int[] nums, int target) {
3          int left = 0;
4          int right = nums.length - 1;
5          while(left <= right) {
6              int mid = left + (right - left) / 2;
7              if(nums[mid] > target) {
8                  right = mid - 1;
9              } else if(target > nums[mid]) {
10                 left = mid + 1;
11             } else if(target == nums[mid]) {
12                 return mid;
13             }
14         }
15         return -1;
16     }
17 }
```