

# Binary Search

1. 题干目标：搜索符合某条件的值/位置...
2. 解题思路：留一半舍一半 → 直接找到（或逐渐逼近）目标值（可能有时候是留一部分舍一部分，不一定一半。

## 3. Keys:

### 1) 取舍/比较条件的选择和设定

（1）绝对等于（最简单基本）

如：在 sorted array 中找到 3 的位置

（2）相对最好（大多数题目）

如：在 sorted array 中找到大于 3 的最小值

### 2) 取舍过后是否需要 post-processing 得到最终值

## 4. 一道基本例题

Find target in sorted array, if exist return index otherwise return -1

\* 红字需思考/注意

### Template 1:

```
public int binarySearch(int [] nums, int target) {  
    if (nums == null || nums.length == 0) { // corner cases  
        return -1;  
    }  
    int left = 0, right = nums.length, mid = 0;  
    while (left <= right) {  
        mid = left + (right - left) / 2; // why not mid = (left + right) / 2  
        if (nums[mid] == target)        return mid;  
        else if (nums[mid] < target)    left = mid + 1;  
        else    right = mid - 1;  
    }  
    return -1;  
}
```

search 部分过后，若仍未找到目标，则 left 和 right 的相对位置是 (right, left)，无需 post-processing

Template 2:

```
public int binarySearch(int [] nums, int target) {
    if (nums == null || nums.length == 0) { // corner cases
        return -1;
    }
    int left = 0, right = nums.length, mid = 0;
    while (left < right) {
        mid = left + (right - left) / 2;
        if (nums[mid] == target)      return mid;
        else if (nums[mid] < target)   left = mid + 1;
        else    right = mid;
    }
    return nums[left] == target ? left : -1;
}
```

search 部分过后，若仍未找到目标，则 left 和 right 的相对位置是 (left = right)，post-processing 还需要检查一个元素

Template 3:

```
public int binarySearch(int [] nums, int target) {
    if (nums == null || nums.length == 0) { // corner cases
        return -1;
    }
    int left = 0, right = nums.length, mid = 0;
    while (left + 1 < right) {
        mid = left + (right - left) / 2;
        if (nums[mid] == target)      return mid;
        else if (nums[mid] < target)   left = mid;
        else    right = mid;
    }
    // 有些题目中先比较 left 还是 right may matters
    if (nums[left] == target)      return left;
    if (nums[right] == target)     return right;
    return -1;
}
```

search 部分过后，若仍未找到目标，则 left 和 right 的相对位置是 (left, right)，post-processing 还需要检查两个元素

思考：left 和 right 每次等于 mid 还是等于  $mid + 1$  或  $mid - 1$  在每个条件下是怎么定的，可以改吗？

前五题：

L278, L35, L162, L34, L74

后五题：

L240, L33, L81, L153, L154