



链滴

算法 02 二分查找

作者: [AshShawn](#)

原文链接: <https://ld246.com/article/1646124394699>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

二分查找多用于有序的数组,时间复杂度为 $O(N)$

1.有序数组查找某数

```
private int binarySearch(int[] arr, int value) {
    //arr非空,长度判断省略
    int mid, L = 0;
    int R = arr.length - 1;
    while (L < R) {
        //计算mid, 这样写比(L+R)/2好,因为L+R时会数据溢出
        mid = L + ((R - L) >> 1);
        if (arr[mid] == value) {
            return mid;
        } else if (arr[mid] < value) {
            L = mid + 1;
        } else {
            R = mid - 1;
        }
    }
    //L==R时
    if (arr[L] == value) {
        return L;
    }
    return -1;
}
```

2.有序数组大于等于某个数的最左位置

```
/**
 * 二分查找 大于等于 某个数的最左index
 */
private int binarySearchLeft(int[] arr, int value) {
    //arr非空,长度判断省略
    int mid, L = 0;
    int R = arr.length - 1;
    int index = -1;
    while (L <= R) {
        //计算mid, 这样写比(L+R)/2好,因为L+R时会数据溢出
        mid = L + ((R - L) >> 1);
        if (arr[mid] >= value) {
            index = mid;
            R = mid - 1;
        } else {
            L = mid + 1;
        }
    }
    return index;
}
```