



链滴

算法 - 高级二分查找

作者: [amoslam](#)

原文链接: <https://ld246.com/article/1578059030491>

来源网站: [链滴](#)

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

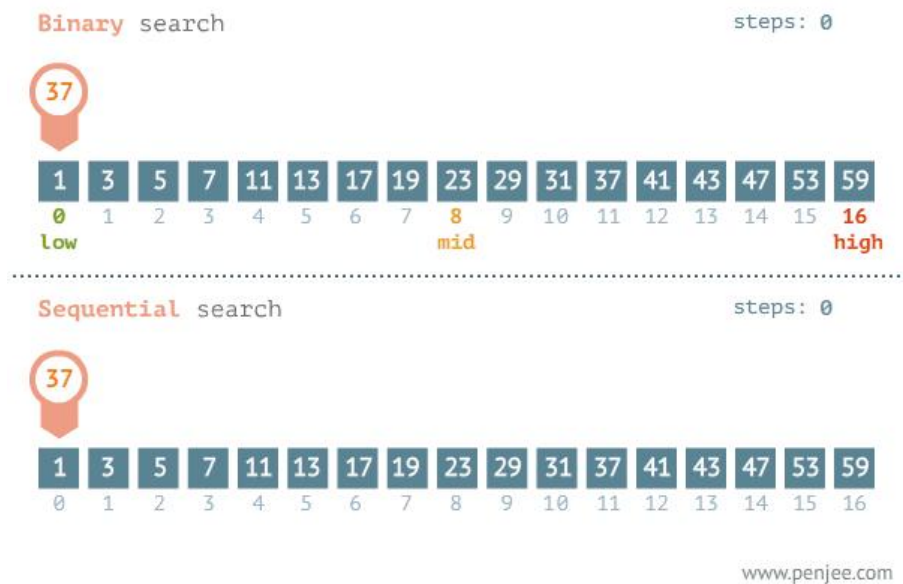
概述

二分查找 (binary search)，也称折半搜索，是一种在 **有序数组** 中 **查找某一特定元素** 的搜索算法。搜索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，则搜索过程结束；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。这种搜索算法每一次比较都使搜索范围缩小一半。

- **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 $O(\log n)$ 。（ n 代表集合中元素的个数）
- **空间复杂度:** $O(1)$ 。虽以递归形式定义，但是尾递归，可改写为循环。

动图演示

图片来源：百度



优缺点

优点

查找速度快，平均性能好

缺点

必须为 **有序数组**

使用场景

1. 优惠券

优惠券最优算法，最优惠的券降序排列，同等优惠的券按覆盖范围优先级升序

- 券使用范围

- 单品券
- 品类券
- 品牌券

思路

- 1.获取用户可用券
- 2.从redis筛选出当前可用券的使用范围
- 3.根据购买的商品检索（二分查找）是否命中
- 4.计算命中券的优惠额度
- 5.快速排序算法降序

公式

- 算法一： $mid = (low + high) / 2$
- 算法二： $mid = low + (high - low) / 2$

代码

```
/**
 * =====
 * 作者: amos lam
 * 时间: 2020年1月3日下午9:39:09
 * 备注: 高级二分查找
 * =====
 */
public class BinarySearch {

    public static int binarySearch(int[] a, int num) {

        int len = a.length;
        int high = len - 1;
        int low = 0;
        while (low <= high) {

            int mid = (high + low) >> 1;
            if (a[mid] == num) {

                return mid;
            }else if (a[mid] < num) {

                low = mid + 1;
            }else {

                high = mid - 1;
            }
        }
    }
}
```

```
        return -1;
    }

    public static void main(String[] args) {

        int[] a = {1,10,30,34,40,45,59};
        int index = binarySearch(a, 30);

        System.out.println(index);
    }
}
```

公共包代码

调用 java.util.Arrays 包的 binarySearch 方法

```
import java.util.Arrays;

/**
 * =====
 * 作者: amos lam
 * 时间: 2020年1月3日下午9:39:09
 * 备注: 高级二分查找
 * =====
 */
public class BinarySearch {

    public static void main(String[] args) {

        int[] a = {1,10,30,34,40,45,59};
        int index = Arrays.binarySearch(a, 30);

        System.out.println(index);
    }
}
```