

# 插值查找

## 【思想】

插值查找是二分查找的改进，插值查根据待查找值自适应地划分有序序列。而不是简单地折半地方式划分序列。

比如要在取值范围1 ~ 10000 之间 100 个元素从小到大均匀分布的数组中查找5，我们自然会考虑从数组下标较小的开始查找，划分序列的时候将前半段的序列长度尽量地小一点，间接地减少比较次数。

经过以上分析，折半查找这种查找方式，不是自适应的（也就是说傻瓜式的）。二分查找中查找点计算如下：

$mid = (low + high) / 2$ , 即  $mid = low + 1/2 * (high - low)$ ;

通过类比，我们可以将查找的点改进为如下：

$mid = low + (key - a[low]) / (a[high] - a[low]) * (high - low)$ ,

也就是将上述的比例参数1/2改进为自适应的，根据关键字在整个有序表中所处的位置，让mid值的变化更靠近关键字key，这样也就间接地减少了比较次数。

## 【复杂度】

对于表长较大，而关键字分布又比较均匀地情况，插值查找比折半查找要好很多。反之，数组中如果分布非常不均匀，那么插值查找未必是很合适的选择。

查找成功或者查找失败复杂度均为  $O(\log_2(\log_2 n))$

## 【代码】

```
int InsertionSearch(int a[], int value, int low, int high)
{
    int mid = low + (value - a[low]) / (a[high] - a[low]) * (high - low);
    if(a[mid] == value)
        return mid;
    if(a[mid] > value)
        return InsertionSearch(a, value, low, mid - 1);
    if(a[mid] < value)
        return InsertionSearch(a, value, mid + 1, high);
}
```