

“ 二分

二分查找

二分查找有 64 种写法。

对其进行分类：
取数方式：向下取整，向上取整（共2种）
区间开闭：闭区间，左闭右开区间，左开右闭区间，开区间（共4种）
问题类型：
①对于不下降序列a，求最小的i，使得a[i]=key
②对于不下降序列a，求最大的i，使得a[i]=key
③对于不下降序列a，求最小的i，使得a[i]>key
④对于不下降序列a，求最大的i，使得a[i]<key
⑤对于不上升序列a，求最小的i，使得a[i]=key
⑥对于不上升序列a，求最大的i，使得a[i]=key
⑦对于不上升序列a，求最小的i，使得a[i]<key
⑧对于不上升序列a，求最大的i，使得a[i]>key（共8种）

- 综上所述，二分查找共有 64 种写法。

对于不下降序列 a，n 为序列 a 元素的个数，key 为关键字。

1. 求最小的 i，使得 a[i]=key，若不存在，则返回 - 1

```
int search1(int a[],int n,int key){
    int m,l=0,r=n-1;
    while(l<r){
        m=l+((r-l)>>1);
        if(a[m]<key)l=m+1;
        else r=m;
    }
    if(a[r]==key)return r;
    return -1;
}
```

2. 求最大的 i，使得 a[i]=key，若不存在，则返回 - 1

```
int search2(int a[],int n,int key){
    int m,l=0,r=n-1;
    while(l<r){
        m=l+((r+1-l)>>1);
        if(a[m]<=key)l=m;
        else r=m-1;
    }
    if(a[l]==key)return l;
    return -1;
}
```

3. 求最小的 i，使得 a[i]>key，若不存在，则返回 - 1

```
int search3(int a[],int n,int key){
    int m,l=0,r=n-1;
    while(l<r){
        m=l+((r-l)>>1);
        if(a[m]<=key)l=m+1;
        else r=m;
    }
    if(a[r]>key)return r;
    return -1;
}
```

4. 求最大的 i，使得 a[i]<key，若不存在，则返回 - 1

```
int search4(int a[],int n,int key){
    int m,l=0,r=n-1;
    while(l<r){
        m=l+((r+1-l)>>1);
        if(a[m]<key)l=m;
        else r=m-1;
    }
    if(a[l]<key)return l;
    return -1;
}
```

- 对于 3、4，也可以先判断是否存在，再进行二分查找。

全文完

本文由 简悦 Simpread 优化，用以提升阅读体验
使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

