

[寻找自我的博客](#)

[编程珠玑第四章 编写正确的程序](#)

2012-08-27 20:36 61人阅读 [评论\(0\)](#) [收藏](#) [举报](#)

1. 二分查找 每次都去掉一半的可能性。每个分支都是等概率的，是平衡的，有最优下界。

2. 在有相同数的数组中有二分查找查出 数第一次出现的位置

```
#include <iostream>
using namespace std;

int bs(int *a, int begin, int end, int v)
{
    int *b = a + begin; //开始
        int *e = a + end; //结束
        int *mid = NULL; //中间

    while (b < e) //直到等于第一个出现的值
    {
        mid = b + ((e - b)>>1); //得到中间位置的地址

        if (*mid >= v)
            e = mid;
        else
            b = mid + 1;
    }
    if ((e-a) < end && (*e == v))
        return e-a;

    return -1;
}

int main()
{
    int a[5]={1,2,2,2,4};
    cout<<bs(a, 0, 4, 2)<<endl;
    return 0;
}
```

6.

每次因为 都要丢掉一颗出去，所以过程会终止。

白色的豆子 有两种情况： 1. 颜色相同 丢2颗 2. 颜色不相同 不丢。

所以 如果白色的豆子是奇数，那就能留下来，否则留下来的是黑色的 。

11. 二分查找递归。

这里给递归和迭代。

```
/**
    *在有序数组中查找，利用二分查找的方法
```

```

        *
        */
#include <iostream>
using namespace std;

/**
    *二分查找
    *
    */
int sort(int *a, int length, int value){
    int left = 0, right = length - 1;
    while(left <= right){
        int center = (left + right)/2;
        if(value < a[center]){
            right = center - 1;
        }else if(value > a[center]){
            left = center + 1;
        }else{
            return center;
        }
    }
    return -1;
}

//递归形式
int sort(int *a, int left, int right, int value){
    int center = (left + right)/2;
    //异常时的检测
    if(left == right && a[center] != value) return -1;
    //递归查找
    if(value > a[center]) sort(a, center+1, right, value);
    else if(value < a[center]) sort(a, left, center-1, value);
    else return center;
}

int main(){
    int a[10] = {2,4,5,6,8,12,32,45,55,65};
    cout<<"列表: "<<endl;
    for(int i=0; i<10; i++){
        cout<<a[i]<<" ";
    }
    cout<<endl;
    int v;
    cin>>v;
    cout<<"在位置:"<<sort(a, 10, v)<<endl;
    cout<<"递归在位置:"<<sort(a, 0,9, v)<<endl;
}

```