

算法设计与分析实验报告

实验名称：二分检索(分治法)

一、问题陈述，相关背景、应用及研究现状的综述分析

1.问题陈述：

给定递增整数序列 L ，其大小为 n ，要求使用二分查找法查找任意元素的位置 k (序列中第几个)

2.相关背景：

在计算机科学中，二分搜索（英语：binary search），也称折半搜索（英语：half-interval search）、对数搜索（英语：logarithmic search），是一种在有序数组中查找某一特定元素的搜索算法。

二、模型拟制、算法设计和正确性证明

1. 算法设计：

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。

三、时间和空间复杂性分析

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：

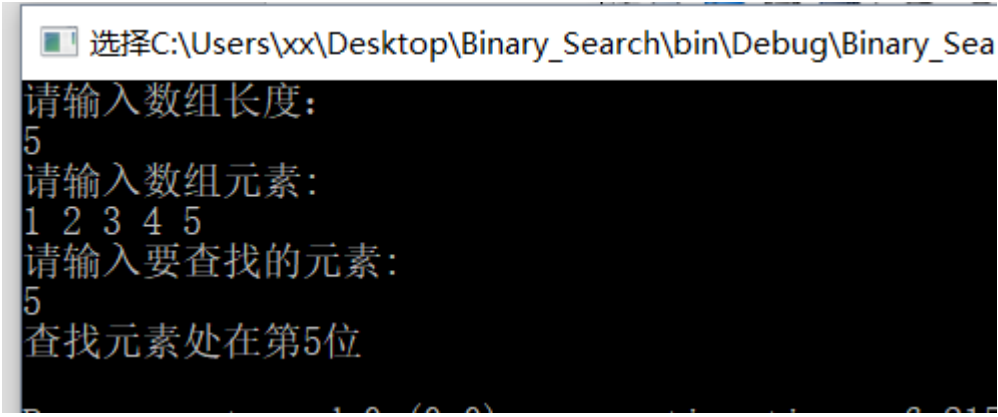
没有使用额外空间，空间复杂度为 $O(1)$

四、程序实现和实验测试过程

源程序：

```
#include <iostream>
using namespace std;
int Binary_Search(int a[],int array_length,int target_element)
{
    int left=0,right=array_length-1;
    while(left<=right)
    {
        int middle =(left+right)/2;
        if(target_element==a[middle])
            return middle+1;
        else if(target_element>a[middle])
            left=middle+1;
        else
            right=middle-1;
    }
    return -1;
}
int main()
{
    int array_length;
    cout<<"请输入数组长度："<<endl;
    cin>>array_length;
    int a[array_length];
    cout<<"请输入数组元素："<<endl;
    for(int i=0;i<array_length;i++)
        cin>>a[i];
    int target_element;
    cout<<"请输入要查找的元素："<<endl;
    cin>>target_element;
    int result=Binary_Search(a,array_length,target_element);
    if(result>0)
        cout<<"查找元素处在第"<<result<<"位"<<endl;
    else
        cout<<"查找元素不存在"<<endl;
    return 0;
}
```

结果：


<h3>五、总结</h3> <p>二分搜索基于分治法，时间复杂度较低。</p>