



***Liaoning Normal University***

**算法设计分析实验报告**

**课程名称：计算机算法设计分析**

**专业：计算机科学与技术（师范）**

**班级： （1） 班**

**学号：201621012299**

**姓名：周惠馨**

**指导教师：王相海**

**2018年11月6日**

1. **问题的提出**

设a[0:n-1]是已排好序的数组，请改写二分搜索算法，使得当x不在数组中时，返回小于x的最大元素位置i和大于x的最小元素位置j。当搜索元素在数组中时、i和j相同，均为x在数组中的位置。

1. **算法的基本思想**

二分搜索方法充分利用了元素间的次序关系，采用分治策略，可在最坏情况下用O(logn)时间完成搜索任务。二分搜索算法的基本思想是将n个元素分成个数大致相同的两半，取a[n/2]与x作比较。如果x=a[n/2]，则找到x，算法终止；如果x<a[n/2]，则只在数组a的左半部继续搜索x；如果x>a[n/2]，则只在数组a的右半部继续搜索x。

1. **算法的程序实现**

#include <bits/stdc++.h>

using namespace std;

const int MAX = 100;

void BinarySearch\_Changed(int \*a, int value, int left, int right, int &i, int &j)

{

int mid, rright = right, lleft = left;

while(left <= right)

{

mid = (left + right) / 2;

if(value == a[mid])

{

i = j = mid;

return;

}

else if(value < a[mid])

{

right = mid - 1;

}

else

{

left = mid + 1;

}

}

i = left - 1, j = left;

if(left > rright)

i = rright, j = -1;

if(right < 0)

i = -1, j = 0;

}

int main()

{

int a[MAX];

int i, j, num, value;

scanf("%d", &num);

for(int i = 0; i < num; ++i)

{

scanf("%d", &a[i]);

}

scanf("%d", &value);

BinarySearch\_Changed(a, value, 0, num - 1, i, j);

printf("i = %d , j = %d", i, j);

}

1. **算法的复杂度分析**

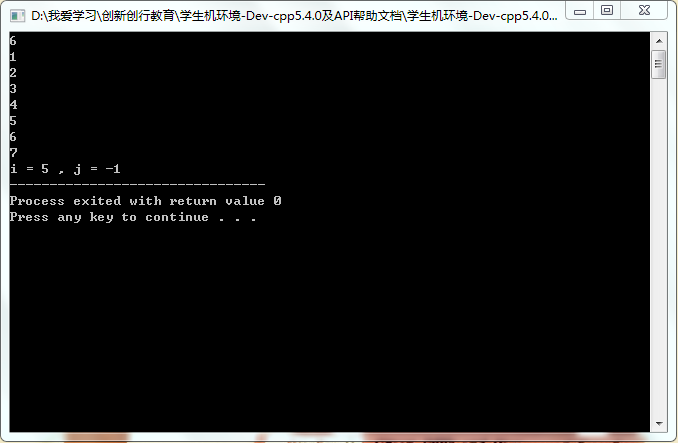
时间复杂度

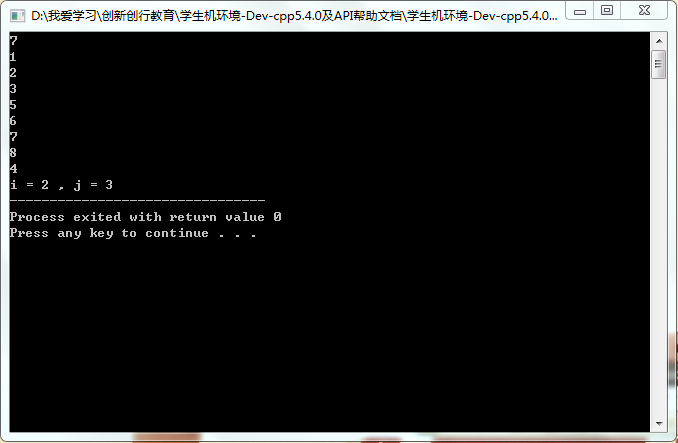
折半搜索每次把搜索区域减少一半，时间复杂度为O（log n）。（n代表集合中元素的个数）

空间复杂度

O（1），虽以递归形式定义，但是尾递归，可改写为循环。

1. **算法运行结果与分析（拷屏黏贴）**





1. **结束语（讨论）**

编写二分查找时可能出现一些问题。

边界错误造成的问题：

二分查找算法的边界，一般来说分两种情况，一种是左闭右开区间，类似于[left,right)，一种是左闭右闭区间，类似于[left,right]，需要注意的是，循环体外的初始化条件，与循环体内的迭代步骤，都必须遵守一致的区间规则，也就是说，如果循环体外初始化时，是以左闭右开区间为边界的，那么循环体内部的迭代也应该如此，如果两者不一致，会造成程序的错误。