实验2 分治法排序查找

# 一. 实验要求

1. 了解用分治法求解的问题的特点和类型；

2. 编程实现典型的分治算法，对算法进行验证分析。

# 二. 实验内容

## 1.准备知识

### （1） 分治法实现二分查找

给定已按升序排好序的n个元素a[0:n-1]，要在这n个元素中找出一特定元素x。



二分查找伪代码如下，建议大家直接根据伪码编写代码，加深对算法理解，尽量不要在直接网上复制。



### （2） 分治法快速排序

**分解**

以*aq*为基准元素将*ap:r*划分成3段*ap:q-1*、*aq*和*aq+1:r*，使得*ap:q-1*中任何元素小于*aq* ，*aq+1:r*中任何元素大于*aq* ；

下标*q*在划分过程中确定。

**递归求解**

递归调用快速排序算法对*ap:q-1*和*aq+1:r*进行排序；

**合并**

对*ap:q-1*和*aq+1:r*的排序是就地进行的，*ap:q-1*和*aq+1:r*排好序后不需要执行任何计算，*ap:r*就已排好序。

快速排序划分算法伪码：



## 2. 实验内容

对乱序整数序列，先用快速排序按非减序排列，再进行二分查找，查找某个元素是否存在，若存在返回匹配的第一个下标位置（从0开始），不存在返回-1。

**示例输入**（第一行表示要查找的数，第二行以后表示数组元素，为编程方便，可限定为20个数，也可不限，自动分配空间）：

95

56

87

789

54

12

35

11

102

456

7821

2

44

245

56

879

45

82

57

62

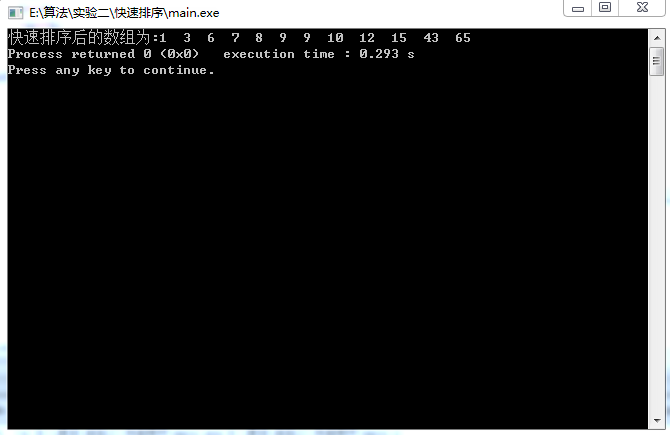
30

**示例输出**（只输出数值即可，不许其他文字）：

-1

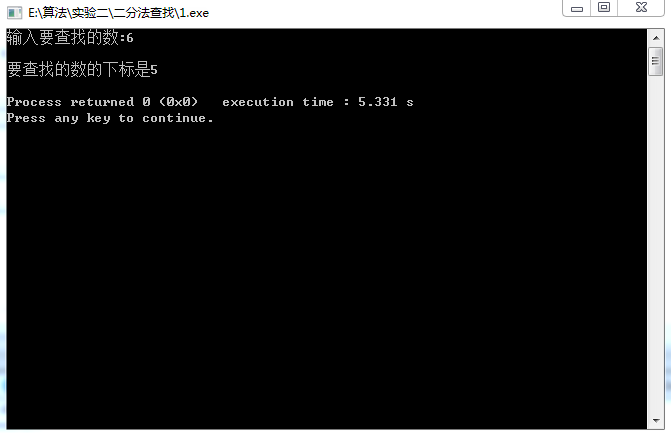
快速排序：





二分查找：





排序查找：

/\*

\*20140031 陈小龙 计科2班

\*/

#include <stdio.h>

#include <stdlib.h>

int swap**(**int **\***a**,**int **\***b**){**

int c**;**

c**=\***a**;**

**\***a**=\***b**;**

**\***b**=**c**;**

**}**

//这里进行快速排序

int **\***Partition**(**int **\***array**,**int l**,**int r**){**

**if(**l**>=**r**)**

**return** array**;**

int p**=**array**[**l**],**i**=**l**,**j**=**r**;**

**while(**i**<**j**){**

**while(**i**<**j **&&** array**[**j**]>=**p**)**

j**--;**

swap**(&**array**[**i**],&**array**[**j**]);**

**while(**i**<**j **&&** array**[**i**]<=**p**)**

i**++;**

swap**(&**array**[**j**],&**array**[**i**]);**

**}**

Partition**(**array**,**l**,**i**-**1**);**

Partition**(**array**,**i**+**1**,**r**);**

**return** array**;**

**}**

//这里进行折半查找

int ersearch**(**int array**[],**int n**,**int k**){**

int low**=**0**,**high**=**n**-**1**,**m**;**

**while(**low**<=**high**){**

m**=(**low**+**high**)/**2**;**

**if(**k**==**array**[**m**])**

**return** m**;**

**else** **if(**k**>**array**[**m**])**

low**=**m**+**1**;**

**else**

high**=**m**-**1**;**

**}**

**return** **-**1**;**

**}**

int main**()**

**{**

int i**=**0**,**a**[]={**56**,**87**,**789**,**54**,**12**,**35**,**11**,**102**,**456**,**7821**,**2**,**44**,**245**,**56**,**879**,**45**,**82**,**57**,**62**,**30**},**k**;**

int **\***b**;**

b**=**Partition**(**a**,**0**,**19**);**

printf**(**"快速排序后的数组为:"**);**

**for(**i**=**0**;**i**<**20**;**i**++){**

printf**(**"%d "**,**b**[**i**]);**

**}**

printf**(**"\n输入要查找的数:"**);**

scanf**(**"%d"**,&**k**);**

printf**(**"该数对应的下标值是：%d\n"**,**ersearch**(**a**,**20**,**k**));**

**return** 0**;**

**}**

