# 习题5 查找

5-1 将关键字2，4，6，8，10，12，14，16依次存放于一维数组A[0...7]中，如果采用折半查找方法查找关键字，在等概率情况下查找成功时的平均查找长度为( A )。

(A) 21/8

(B) 7/2

(C) 4

(D) 9/2

5-2 简单描述静态查找和动态查找的区别。

静态查找: 不改变数据的情况下，进行的查找。

动态查找:在查找的过程中，会改变元素的顺序，可能会进行插入和删除操作。

5-3 设数组A中只存放正数和负数。试设计算法，将A中的负数调整到前半区间，正数调整到后半区间。分析算法的时间复杂度。

#include <iostream>

using namespace std;

void process(int \*a,int n)

{

int left=1;

int right=n;

for(int i=1;i<=n; )

{

if(a[i]<0)

{

left++;

i++;

}

else

{

swap(a[i],a[right]);

right--;

}

if(left==right)

break;

}

}

void swap(int &a,int &b)

{

int temp=a;

a=b;

b=temp;

return ;

}

int main()

{

int a[15]={0};

int n=10;

for(int i=1;i<=n;i++)

{

cin>>a[i];

}

process(a,n);

for(int i=1;i<=n;i++)

cout<<a[i]<<" ";

cout<<endl;

}

算法的时间复杂度为O（n）

5-4 链地址法是Hash表的一种处理冲突的方法，它是将所有哈希地址相同的数据元素都存放在同一个链表中。关于链地址法的叙述，不正确的是( C )。

(A) 平均查找长度较短

(B) 相关查找算法易于实现

(C) 链表的个数不能少于数据元素的个数

(D) 更适合于构造表前无法确定表长的情况

5-5 设哈希(Hash)函数H(k)=(3k)%11，用线性探测再散列法处理冲突，di=i。已知为关键字序列22，41，53，46，30，13，01，67构造哈希表如下：

哈希地址 0 1 2 3 4 5 6 7 8 9 10

关键字 22 41 30 01 53 46 13 67

则在等概率情况下查找成功时的平均查找长度是( A )。

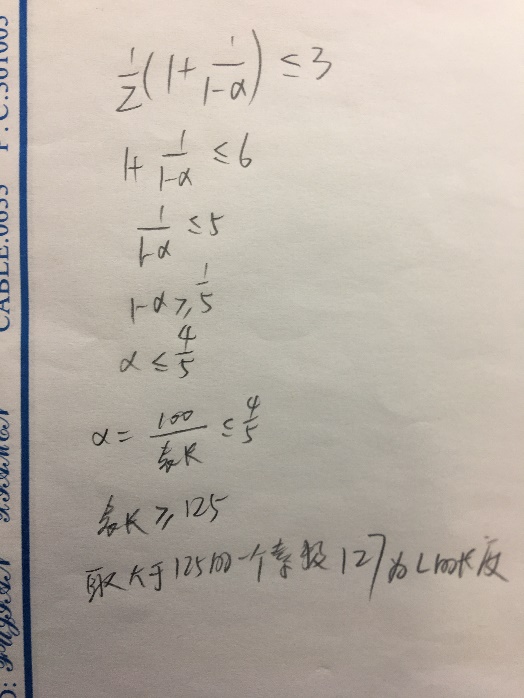
(A) 2

(B) 24/11

(C) 3

(D) 3.5

5-6 有100个不同的关键字拟存放在哈希表L中。处理冲突的方法为线性探测再散列法，其平均查找长度为。试计算L的长度(一个素数)，要求在等概率情况下，查找成功时的平均查找长度不超过3。

素数表：101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167。

5-7 在平衡二叉树中，插入关键字46后得到一颗新的平衡二叉树。在新的平衡二叉树中，关键字37所在结点的左、右孩子结点中保存的关键字是( C )。

(A) 18，46

(B) 25，46

(C) 25，53

(D) 25，69

5-8 用依次插入关键字的方法，为序列{ 5, 4, 2, 8, 6, 9 }构造一棵平衡二叉树(要求分别画出构造过程中的各棵不平衡二叉树)。

