**第5章 查找 习题答案**

**4、习题**

**一、填空题**

1. 顺序查找法，表中元素可以 任意 存放。
2. 在分块查找方法中，首先查找 索引 ，然后再查找相应的块。
3. 顺序查找、二分查找、分块查找都属于 静态 查找。
4. 静态 查找表所含元素个数在查找阶段是固定不变的。
5. 对于长度为n的线性表，若进行顺序查找，则时间复杂度为 O（n） 。
6. 对于长度为n的线性表，若采用二分查找，则时间复杂度为： O（log2n） 。
7. 理想情况下，在散列表中查找一个元素的时间复杂度为： O（1） 。
8. 在关键字序列（7，10，12，18，28，36，45，92）中，用二分查找法查找关键字92，要比较 4 次才找到。
9. 设有100个元素，用二分查找时，最大的比较次数是 7 次。
10. 对二叉排序树进行查找的方法是用待查的值与根结点的键值进行比较，若比根结点小，则继续在 左 子树中查找。
11. 二叉排序树是一种 动态 查找表。
12. 哈希表是按 散 列 存储方式构造的存储结构
13. 哈希法既是一种存储方法，又是一种 查找 方法。
14. 散列表的查找效率主要取决于散列表造表时选取的散列函数和处理 冲突 的方法。
15. 设散列函数H和键值k1，k2，若k1≠k2，且H（k1）=H（k2），则称这种现象为 冲突 。
16. 处理冲突的两类主要方法是开放定址法和 拉链法（或链地址法） 。
17. 散列表（或散列） 查找法的平均查找长度与元素个数n无关。
18. 在哈希函数H(key)=key % P中，P一般应取 质数 。
19. 在查找过程中有插入元素或删除元素操作的，称为 动态 查找。
20. 各结点左右子树深度之差的绝对值至多为 1 的二叉树称谓平衡二叉树。
21. 在10阶B—树中根结点所包含的关键码个数最多为 9 ，最少为 1 。

【分析】m阶的B－树中每个结点至多有m棵子树，若根结点不是终端结点，则至少有两棵子树，每个结点中关键码的个数为子树的个数减1。

1. 一棵5阶B—树中，除根结点外，每个结点的子树树目最少为 3 ，最多为 5 。

【分析】m阶的B－树中每个结点至多有m棵子树，除根结点之外的所有非终端结点至少有?m/2? 棵子树。

1. 对于包含n个关键码的m阶B—树，其最小高度是logm(n+1)，最大高度是logm/2(n+1)/2+1。
2. 在一棵B—树中删除关键码，若最终引起树根结点的合并，则新树比原树的高度减少1层。
3. 在一棵高度为h的B—树中，叶子结点处于第h+1层，当向该B—树中插入一个新关键码时，为查找插入位置需读取 h 个结点。

【分析】B－树的叶子结点可以看作是外部结点（即查找失败）的结点，通常称为外结点。实际上这些结点不存在，指向这些结点的指针为空，B－树将记录插入在终端结点中。

1. 对于长度为n的线性表，若采用分块查找（假定总块数和每块长度均接近 ，用顺序查找确定所在块），则时间复杂性为O(n1/2 )。

**二、选择题**

（1）查找表是以（ A ）为查找结构。

A．集合 B．图 C．树 D．文件

（2）顺序查找法适合于存储结构为（ B ）的线性表。

A．散列存储 B．顺序存储或链接存储

C．压缩存储 D．索引存储

（3）在表长为ｎ的链表中进行线性查找，它的平均查找长度为（ B ）。

Ａ. ASL＝ｎ; Ｂ. ASL＝(n+1)/２;

Ｃ. ASL＝＋１; Ｄ. ASL≈log2ｎ

（4）对线性表进行二分查找时，要求线性表必须 （ D ）。

A．以顺序方式存储 B．以链接方式存储，且结点按关键字有序排序

C．以链接方式存储 D．以顺序方式存储，且结点按关键字有序排序

（5）衡量查找算法效率的主要标准是（ B ）。

A．元素个数 B. 平均查找长度 C．所需的存储量 D．算法难易程度

（6）如果要求一个线性表既能较快地查找，又能适应动态变化的要求，可以采用（ A ）查找方法。

A．分块 B．顺序 C．二分 D．散列

（7） 链表适用于（ A ）查找。

A．顺序 B．二分 C．随机 D．顺序或二分

（8）一个有序表为{1，3，9，12，32，41，45，62，75，77，82，95，100}，当二分查找值为82的结点时，( C )次比较后查找成功。

A．2 B．3 C．4 D．5

（9）二分查找有序表{4，6，10，12，20，30，50，70，88，100},若查找表中元素58，则它将依次与表中（ B ）比较大小，查找结果是失败。

A．30，88，70，50 B．20，70，30，50 C．20，50 D．30，88，50

（10）对有14个元素的有序表A[1**..**14]作二分查找，查找元素A[4]时的被比较元素依次为（ C ）。

A．A[1]，A[2]，A[3]，A[4] B．A[1]，A[14]，A[7]，A[4]

C．A[7]，A[3]，A[5]，A[4] D．A[7]，A[5]，A[3]，A[4]

（11）有一个长度为12的有序表，按二分查找法对其进行查找，在表内各元素等概率情况下查找成功所需的平均比较次数为（ B ）。

A．35／12 B．37／12 C．39／12 D．43／12

（12）采用分块查找时，若线性表共有625个元素，查找每个元素的概率相等，假设采用顺序查找来确定结点所在的块时，每块分（ C ）个结点最佳。

A．6 B．10 C．25 D．625

（13）下列（ C ）不是利用查找表中数据元素的关系进行查找的方法。

A．平衡二叉树 B．有序表的查找

C. 散列查找 D．二叉排序树的查找

（14）设哈希表长m=14，哈希函数H（key）=key％11。表中已有4个结点：

addr(15)=4

addr(38)=5

addr(61)=6

addr(84)=7

其余地址为空。如用二次探测再散列处理冲突，关键字为49的结点的地址是（ D ）。

A．8 B．3 C．5 D．9

（15）对包含n个元素的散列表进行查找，平均查找长度为（ D ）。

A．O(n2) B．O(log2n) C. O(n) D．不直接依赖于n

（16）冲突指的是（ C ）。

A．两个元素具有相同序号 B. 两个元素的键值不同

C．不同键值对应相同的存储地址 D．两个元素的键值相同

（17）在查找过程中，不做增加、删除或修改的查找称为（ A ）。

A．静态查找 B. 内创造 C．动态查找 D．外查找

（18）已知8个元素为{34，76，45，18，26，54，92，65}，按照依次插入结点的方法生成一棵二叉树(不用平衡)，最后两层上结点的总数为（ B ）。

A．1 B．2 C. 3 D．4

（19）不可能生成下图二叉排序树的关键字的序列是（ A ）。

A． 4 5 3 1 2 B．4 2 5 3 1 C．4 5 2 1 3 D．4 2 3 1 5

（20）动态查找包括（ B ）查找。

A．顺序表 B. 二叉排序树

C．有序表 D．索引顺序表

（21）当向一棵m阶的B-树做插入操作时，若一个结点中的关键字个数等于（ A ），则必须分裂为两个结点。

A.m B. m-1 C.m+1 D. m/2

（22）在一个5阶的B-树上，每个非终端结点所含的子树数最少为（ B ）。

A. 2 B.3 C.4 D.5

**三、应用题**

（1） 对于给定结点的关键字集合K={5，7，3，1，9，6，4，8，2，10}，

1）试构造一棵二叉排序树；

2）求等概率情况下的平均查找长度ASL。

解：1）构造二叉排序树 ： 2）ASL=(1\*1+2\*2+3\*4+4\*3)/10=2.9

（2）对于给定结点的关键字集合K={10，18，3，5，19，2，4，9，7，15}，

1）试构造一棵二叉排序树；

2）求等概率情况下的平均查找长度ASL。

解：1）构造二叉排序树：

2）ASL=(1\*1+2\*2+3\*4+4\*2+5\*1)/10=3

（3）将数据序列：25，73，62，191，325，138，依次插入下图所示的二叉排序树，并画出最后结果。

解：

（4）对于给定结点的关键字集合K={1，12，5，8，3，10，7，13，9}，

1）试构造一棵二叉排序树；

2）在二叉树排序BT中删除“12”后的树结构：

或

（5）对于给定结点的关键字集合K={34，76，45，18，26，54，92，38}，

1）试构造一棵二叉排序树；

2）求等概率情况下的平均查找长度ASL。

解：

1）构造二叉排序树

2）ASL=（1\*1+2\*2+3\*3+4\*2）/ 8 =2.75 （或=11/4）

（6）对于给定结点的关键字集合K={4，8，2，9，1，3，6，7，5}，

1）试构造一棵二叉排序树；

2）求等概率情况下的平均查找长度ASL。

解：

1）构造二叉排序树

2）ASL=（1\*1+2\*2+3\*4+4\*2）/ 9 =2.78 （或=25/9）

（7）画出对长度为10的有序表进行折半查找的判定树，并求其等概率时查找成功的平均查找长度。

解：长度为10的判定树：

------------------------------- 1次能找到

------------------------ 2次能找到

------------------ 3次能找到

------------- 4次能找到

ASL=1/10（1\*1+2\*2+3\*4+4\*3）=2.9

（8）二叉排序树如图所示，分别画出：

1）画出删除关键字15以后的二叉树，并要求其平均查找长度尽可能小；

2）在原二叉排序树（即没有删除15）上，插入关键字20。

解：

1） 2）

（9）给定结点的关键字序列为：19，14，23，1，68，20，84，27，55，11，10，79。

设散列表的长度为13，散列函数为：H（K）=K % 13。试画出线性探测再散列解决冲突时所构造的散列表，并求出其平均查找长度。

解：线性探测再散列解决冲突时所构造的散列表：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|  | 14 | 1 | 68 | 27 | 55 | 19 | 20 | 84 | 79 | 23 | 11 | 10 |

① ② ① ④ ③ ① ① ③ ⑨ ① ① ③

平均查找长度ASL=（1\*6+2\*1+3\*3+4\*1+9\*1）/12=30/3=3

（10）给定结点的关键字序列为：47，7，29，11，16，92，22，8，3，哈希表的长度为11。

设散列函数为：H（K）=K % 11。试画出平方探测再散列解决冲突时所构造的散列表，并求出其平均查找长度。

解：平方探测再散列解决冲突时所构造的散列表。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 22 | 3 | 47 | 92 | 16 |  | 7 | 29 | 8 |  |

① ② ② ① ① ① ① ② ②

平均查找长度ASL=（1\*5+2\*4）/9 = 13/9 = 5/3 （或1.44）

（11）给定结点的关键字序列为：19，14，23，1，68，20，84，27，55，11，10，79。

设散列表的长度为13，散列函数为：H（K）=K % 13。试画出链地址法解决冲突时所构造的哈希表，并求出其平均查找长度。

解：链地址法解决冲突时所构造的哈希表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  | 14 |  |  | 1 |  |  | 27 |  |  | 79 | ∧ |
| 2 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  | 68 |  |  | 55 | ∧ |  |  |  |  |  |  |
| 4 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  | 19 |  |  | 84 | ∧ |  |  |  |  |  |  |
| 7 |  |  | 20 | ∧ |  |  |  |  |  |  |  |  |  |
| 8 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  | 23 |  |  | 10 | ∧ |  |  |  |  |  |  |
| 11 |  |  | 11 | ∧ |  |  |  |  |  |  |  |  |  |
| 12 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |

平均查找长度ASL=（1\*6+2\*4+3\*1+4\*1）/12 = 21/12 =7/4 (或1.75)

（12）给定结点的关键字序列为：47，7，29，11，16，92，22，8，3，哈希表的长度为11。

设散列函数为：H（K）=K %11。试画出链地址法解决冲突时所构造的哈希表，并求出其平均查找长度。

解：

链地址法解决冲突时所构造的哈希表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 |  |  | 11 |  |  | 22 | ∧ |  |  |  |  |  |  |
| 1 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  | 47 |  |  | 3 | ∧ |  |  |  |  |  |  |
| 4 |  |  | 92 | ∧ |  |  |  |  |  |  |  |  |  |
| 5 |  |  | 16 | ∧ |  |  |  |  |  |  |  |  |  |
| 6 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  | 7 |  |  | 29 | ∧ |  |  |  |  |  |  |
| 8 |  |  | 8 | ∧ |  |  |  |  |  |  |  |  |  |
| 9 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | ∧ |  |  |  |  |  |  |  |  |  |  |  |  |

平均查找长度ASL=（1\*6+2\*3）/9 = 12/9 = 4/3 （或1.33）

**五．算法设计题**

（1）设单链表的结点是按关键字从小到大排列的，试写出对此链表进行查找的算法。如果查找成功，则返回指向关键字为x的结点的指针，否则返回NULL。

解：//实现本题功能的算法如下，如果查找成功，则返回指向关键字为x的结点的指针，否则返回NULL。

node \*sqsearch(node\*head，int x)

{

node \*p=head;

while(p!=NULL)

{ if(x>p->key)

p=p->link;

else

if(x==p->key)

return p;

else

{ p=NULL;

return p;

}

}

}

（2）试设计一个在用开放地址法解决从突的散列表上删除一个指定结点的算法。

解：//算法思想是：首先计算要删除的关键字为k的记录所在的位置，将其置为空(即删除)，然后利用线性探测法查找是否有与k发生冲突而存储到下一地址的记录，如果有则将记录移到原来k所在的位置，直至表中没有与k冲突的记录为止。实现本题功能的算法如下：

void delete(sqlist r，int n，int k)

{ int h,h0,h1;

h=k%n;

while(r[h].key!=k)

h=(h+1)％n;

r[h]=NULL;

h0=h;

hl=(h+1)％n;

while(hl!=h)

{

while(r[h1].key％n!=h)

hl=(hl+1)％n;

r[h0]=r[h1];

r[h1]=NULL;

h0=h1;

h1=(hl+1)％n;

}

}

（3）设给定的散列表存储空间为H[1-m]，每个单元可存放一个记录，H[i]的初始值为零，选取散列函数为H(R.key)，其中key为记录R的关键字，解决冲突的方法为线性控测法，编写一个函数将某记录R填入到散列表H中。

解：//本题的算法思想：先计算地址H(R.key)，如果没有冲突，则直接填入;否则利用线性探测法求出下一地址，直到找到一个为零的地址，然后填入。实现本题功能的函数如下：

void insert(record H,int m,record R)

{ int i;

i=H(R.key);

if(H[i]==NULL)

H[i]=R;

else

{

while (H[i]!=NULL)

i=(i+1)%(m+1);

H[i]=R;

}

}