第9章 集合

一．选择题

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.C | 2.A | 3.1D | 3.2C | 4.D | 5.B | 6.D | 7.D | 8.C | 9.A | 10.D | 11.B |
| 12.1C | 12.2C | 13.1C | 13.2D | 13.3G | 13.4H | 14.1E | 14.2B | 14.3E | 14.4B | 14.5B | 15.1B |
| 15.2A | 16.A | 17.C | 18.C | 19.C | 20.D | 21.B | 22.C | 23.B | 24.C | 25.1B | 25.2F |
| 25.3I | 26.A | 27.D | 28.C | 29.1A | 29.2C | 30.B | 31.D | 32.D | 33.C | 34.D | 35.1D |
| 35.2C | 36.C |  |  |  |  |  |  |  |  |  |  |

二．判断题

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.√ | 2.√ | 3.× | 4.× | 5.× | 6.√ | 7.√ | 8.× | 9.× | 10.× | 11.× | 12.√ |
| 13.√ | 14.× | 15.× | 16.× | 17.√ | 18.× | 19.√ | 20.× | 21.× | 22.× | 23.× | 24.× |
| 25.√ | 26.× | 27.× | 28.√ | 29.√ | 30.× | 31.× | 32.√ | 33.√ | 34.× | 35.√ | 36.√ |

部分答案解释如下。

4．不能说哪种哈希函数的选取方法最好，各种选取方法有自己的适用范围。

8．哈希表的结点中可以包括指针，指向其元素。

11．单链表不能使用折半查找方法。

20．按插入后中序遍历是递增序列的原则，若某结点只有右子树，而插入元素的关键字小于该结点的关键字，则会插入到该结点的左侧，成为其左孩子。这种插入就不是插入到叶子下面。

21．从平衡因子定义看，完全二叉树任一结点的平衡因子的绝对值确实是小于等于1。但是，平衡二叉树本质上是二叉排序树，完全二叉树不一定是排序树。故不能说完全二叉树是平衡二叉树。

23．某结点的左子树根结点不一定是它的中序前驱，其右子树根结点也不一定是它的中序后继。

24．在等概率下，查找成功时的平均查找长度相同，查找失败时的平均查找长度不相同。

26．只有被删除结点是叶子结点时命题才正确。

三．填空题

1．n n+1 2．4 3．6,9,11,12 4．5

5．26（第4层是叶子结点，每个结点两个关键字） 6．1,3,6,8,11,13,16,19

7．5,96 8．m-1,「m/2⎤-1 9．2,4,3

10．(1)哈希函数(2)解决冲突的方法 (3)选择好的哈希函数 (4)处理冲突的方法 (5)均匀(6)简单

11．AVL树（高度平衡树，高度平衡的二叉排序树），或为空二叉树，或二叉树中任意结点左子树高度与右子树高度差的绝对值小于等于1。

12．小于等于表长的最大素数或不包含小于20的质因子的合数 13．16 14．⎣㏒2n」+1

15．(1)45 (2)45 (3)46（块内顺序查找） 16．k(k+1)/2 17．30，31.5（块内顺序查找）

18．(1)顺序存储或链式存储 (2)顺序存储且有序 (3)块内顺序存储，块间有序 (4) 散列存储

19．(n+1)/2 20．(n+1)/n\*log2(n+1)-1 21．结点的左子树的高度减去结点的右子树的高度

22．(1)顺序表(2)树表(3)哈希表(4)开放定址方法(5)链地址方法(6)再哈希(7)建立公共溢出区

23．直接定址法 24．log⎡m/2⎤()+1 25．O(N) 26．n(n+1)/2

27．54 28．31 29．37/12 30．主关键字 31．左子树 右子树

32．插入 删除 33．14 34．(1)126 (2)64 (3)33 (4)65

35．（1）low<=high (2) (low+hig) DIV 2 (3) binsrch:=mid (4)binsrch:=0

36．(1) k (2) I<n+1 37．(1)rear=mid-1 (2)head=mid+1 (3)head>rear

38．(1)p!=null (2)pf=p (3)p!=\*t (4)\*t=null

四．应用题

1．概念是基本知识的主要部分，要牢固掌握。这里只列出一部分，目的是引起重视，解答略。

2．（1）散列表存储的基本思想是用关键字的值决定数据元素的存储地址

（2）散列表存储中解决碰撞的基本方法：

① 开放定址法 形成地址序列的公式是：Hi=（H（key）+di）% m，其中m是表长，di是增量。根据di取法不同，又分为三种：

a．di =1，2，…，m-1 称为线性探测再散列，其特点是逐个探测表空间，只要散列表中有空闲空间，就可解决碰撞，缺点是容易造成“聚集”，即不是同义词的关键字争夺同一散列地址。

b．di =12，-12，22，-22，… ，±k2（k≤m/2） 称为二次探测再散列，它减少了聚集，但不容易探测到全部表空间，只有当表长为形如4j+3（j为整数）的素数时才有可能。

c．di =伪随机数序列，称为随机探测再散列。

② 再散列法 Hi=RHi（key） i=1，2，…，k，是不同的散列函数，即在同义词产生碰撞时，用另一散列函数计算散列地址，直到解决碰撞。该方法不易产生“聚集”，但增加了计算时间。

③ 链地址法 将关键字为同义词的记录存储在同一链表中，散列表地址区间用H[0..m-1]表示，分量初始值为空指针。凡散列地址为i（0≤i≤m-1）的记录均插在以H[i]为头指针的链表中。这种解决方法中数据元素个数不受表长限制，插入和删除操作方便，但增加了指针的空间开销。这种散列表常称为开散列表，而①中的散列表称闭散列表，含义是元素个数受表长限制。

④ 建立公共溢出区 设H[0..m-1]为基本表，凡关键字为同义词的记录，都填入溢出区

O[0..m-1]。

（3）用分离的同义词表和结合的同义词表解决碰撞均属于链地址法。链地址向量空间中的每个元素不是简单的地址，而是关键字和指针两个域，散列地址为i（0≤i≤m-1）的第一个关键字存储在地址空间向量第i个分量的“关键字”域。前者的指针域是动态指针，指向同义词的链表，具有上面③的优缺点；后者实际是静态链表，同义词存在同一地址向量空间（从最后向前找空闲单元），以指针相连。节省了空间，但易产生“堆积”，查找效率低。

（4）要在被删除结点的散列地址处作标记，不能物理的删除。否则，中断了查找通路。

（5）记录 负载因子

3．评价哈希函数优劣的因素有：能否将关键字均匀影射到哈希空间上，有无好的解决冲突的方法，计算哈希函数是否简单高效。由于哈希函数是压缩映像，冲突难以避免。解决冲突的方法见上面2题。

4．哈希方法的平均查找路长主要取决于负载因子（表中实有元素数与表长之比），它反映了哈希表的装满程度，该值一般取0.65~0.9。解决冲突方法见上面2题。

5．不一定相邻。哈希地址为i（0≤i≤m-1）的关键字，和为解决冲突形成的探测序列i的同义词，都争夺哈希地址i。

6．

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | 14 | 01 | 9 | 23 | 84 | 27 | 55 | 20 |  |  |
| 比较次数 | 1 | 1 | 1 | 2 | 3 | 4 | 1 | 2 |  |  |

平均查找长度：ASLsucc=（1+1+1+2+3+4+1+2）/8=15/8

以关键字27为例：H（27）=27%7=6（冲突） H1=（6+1）%10=7（冲突）

H2=（6+22）%10=0（冲突） H3=（6+33）%10=5 所以比较了4次。

7．由于装填因子为0.8，关键字有8个，所以表长为8/0.8=10。

（1）用除留余数法，哈希函数为H（key）=key % 7

（2）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | 21 | 15 | 30 | 36 | 25 | 40 | 26 | 37 |  |  |
| 比较次数 | 1 | 1 | 1 | 3 | 1 | 1 | 2 | 6 |  |  |

（3）计算查找失败时的平均查找长度，必须计算不在表中的关键字，当其哈希地址为i（0≤i≤m-1）时的查找次数。本例中m=10。故查找失败时的平均查找长度为：

ASLunsucc=（9+8+7+6+5+4+3+2+1+1）/10=4.6 ASLsucc =16/8=2

（4）**int**  Delete（**int**  h[n]，**int**  k）

// 从哈希表h[n]中删除元素k，若删除成功返回1，否则返回0

{i=k%7；// 哈希函数用上面（1），即H（key）=key % 7

**if**（h[i]== maxint）//maxint解释成空地址

printf（“无关键字%d\n”，k）；**return** （0）；}

**if**（h[i]==k）{h[i]=-max**int** ；**return** （1）；} //被删元素换成最大机器数的负数

**else** // 采用线性探测再散列解决冲突

{j=i;

**for**（d=1；d≤n-1；d++）

{i=（j+d）%n； // n为表长，此处为10

**if**（h[i]== maxint）return (0); //maxint解释成空地址

**if**（h[i]==k）{ h[i]=-maxint；**return** （1）；}

}//**for**

**}**

printf（“无关键字%d\n”，k）；return (0)

}

8．

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 |  | 15 |  | 24 | 10 | 19 | 17 | 38 | 18 | 40 |
| 比较次数 |  | 1 |  | 1 | 2 | 1 | 4 | 5 | 5 | 5 |

哈希表a: ASLsucc=24/8=3；

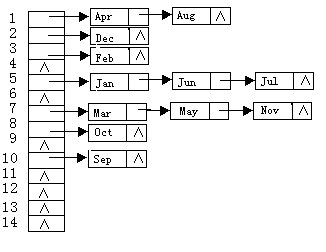
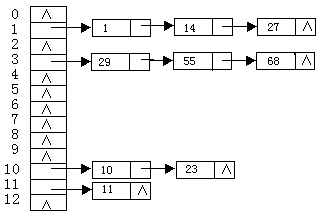
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 |  | 15 | 17 | 24 | 10 | 19 | 40 | 38 | 18 |  |
| 比较次数 |  | 1 | 3 | 1 | 2 | 1 | 2 | 4 | 4 |  |

哈希表b: ASLsucc =18/8

9．（1）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 关键字 | 13 | 22 |  | 53 | 1 |  | 41 | 67 | 46 |  | 51 |  | 30 |
| 比较次数 | 1 | 1 |  | 1 | 2 |  | 1 | 2 | 1 |  | 1 |  | 1 |

（2）装填因子=9/13=0.7 （3）ASLsucc =11/9 （4）ASLunsucc =29/13

10． 11．ASLsucc=19/12

12．常用构造哈希函数的方法有：

（1）数字分析法 该法事先需知道关键字集合，且关键字位数比散列表地址位数多，应选数字分布均匀的位。

（2）平方取中法 将关键字值的平方取中间几位作哈希地址。

（3）除留余数法 H（key）=key%p，通常p取小于等于表长的最大素数。

（4）折叠法 将关键字分成长度相等（最后一段可不等）的几部分，进行移位叠加或间界叠加，其值作哈希地址。

（5）基数转换法 两基数要互素，且后一基数要大于前一基数。

在哈希表中删除一个记录，在拉链法情况下可以物理地删除。在开放定址法下，不能物理地删除，只能作删除标记。该地址可能是该记录的同义词查找路径上的地址，物理的删除就中断了查找路径。因为查找时碰到空地址就认为是查找失败。

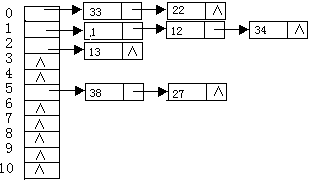
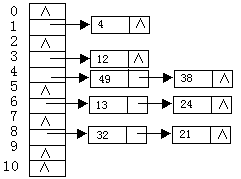
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 关键字 |  | 14 | 01 | 68 | 27 | 55 | 19 | 20 | 84 | 79 | 23 | 11 | 10 |  |  |  |
| 比较次数 |  | 1 | 2 | 1 | 4 | 3 | 1 | 1 | 3 | 9 | 1 | 1 | 3 |  |  |  |

13．（1）

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 关键字 |  | 4 |  | 12 | 49 | 38 | 13 | 24 | 32 | 21 |  |
| 比较次数 |  | 1 |  | 1 | 1 | 2 | 1 | 2 | 1 | 2 |  |

ASLsucc =（1+1+1+2+1+2+1+2）/8=11/8

ASLunsucc=（1+2+1+8+7+6+5+4+3+2+1）/11=40/11

（2）

13题图ASLsucc =11/8 ASLunsucc=19/11 14题（2） ASLsucc=13/8 ASLunsucc=19/11

值得指出，对用拉链法求查找失败时的平均查找长度有两种观点。其一，认为比较到空指针算失败。以本题为例，哈希地址0、2、5、7、9和10均为比较1次失败，而哈希地址1和3比较2次失败，其余哈希地址均为比较3次失败，因此，查找失败时的平均查找长度为19/11，我们持这种观点。还有另一种理解，他们认为只有和关键字比较才计算比较次数，而和空指针比较不计算。照这种观点，本题的ASLunsucc=（1+1+2+2+2）/11=8/11

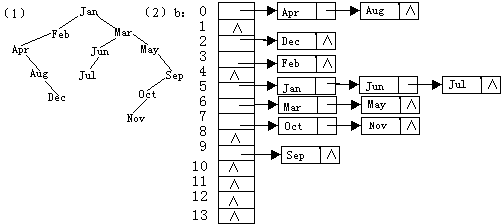
14．由hashf(x)=x mod 11 可知，散列地址空间是0到10，由于有8个数据，装载因子取0.7。

（1）

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 关键字 | 33 | 1 | 13 | 12 | 34 | 38 | 27 | 22 |  |  |  |
| 比较次数 | 1 | 1 | 1 | 3 | 4 | 1 | 2 | 8 |  |  |  |

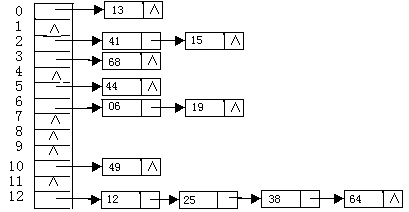
ASLsucc=21/8 ASLunsucc=47/11

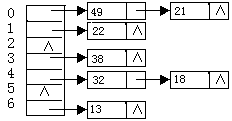
15．

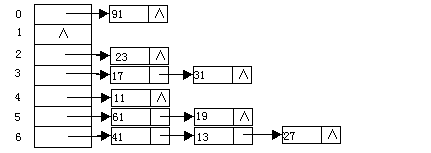
（1）ASL=42/12 

 （2）a：ASLsucc=31/12 （2）b：ASLsucc=18/12 (注：本题[x]取小于等于x的最大整数)

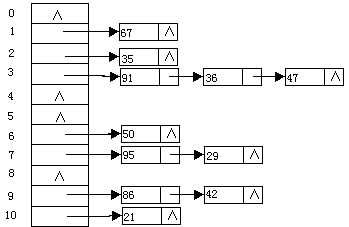
16．

17．查找时，对关键字49,22,38,32,13各比较一次，对21,18各比较两次



18．ASLsucc =15/10

19．ASLsuss =16/11



20．

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 关键字 | 231 | 89 | 79 | 25 | 47 | 16 | 38 | 82 | 51 | 39 | 151 |  |
| 比较次数 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 3 | 2 | 4 | 3 |  |

ASLsucc =21/11

21．

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 关键字 | 22 | 33 | 46 | 13 | 01 | 67 |  |  | 41 | 53 | 30 |
| 比较次数 | 1 | 2 | 1 | 2 | 4 | 5 |  |  | 1 | 1 | 3 |

22．

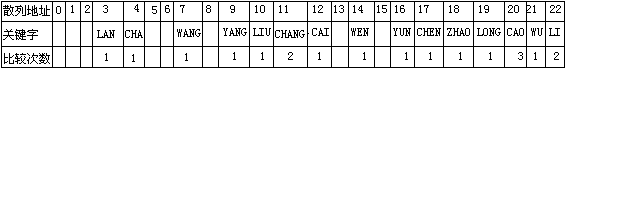
（1）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 关键字 | 32 | 17 | 63 | 49 |  |  |  |  | 24 | 40 | 10 |  |  |  | 30 | 31 | 46 | 47 |
| 比较次数 | 1 | 1 | 6 | 3 |  |  |  |  | 1 | 2 | 1 |  |  |  | 1 | 1 | 3 | 3 |

（2）查找关键字63，H（k）=63 MOD 16=15，依次与31，46，47，32，17，63比较。

（3）查找关键字60，H（k）=60 MOD 16=12，散列地址12内为空，查找失败。

（4）ASLsucc=23/11

23．设用线性探测再散列解决冲突，根据公式Snl≈（1+1/（1-α）） /2 。可求出负载因子为α=0.67。再根据数据个数和装载因子，可求出表长m=15/0.67，取m=23。设哈希函数H（key）=（关键字首尾字母在字母表中序号之和）MOD 23。

从上表求出查找成功时的平均查找长度为ASLsucc=19/15<2.0，满足要求。

24．（1）哈希函数H（key）=（关键字各字符编码之和）MOD 7

（2）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | be | cd | aa | ab | ac | ad | bd | bc | ae | ce |
| 比较次数 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 9 |

25．α=0.7，所以表长取m=7/0.7=10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | SAT |  | WED |  |  | SUN | MON | TUE | THU | FRI |
| 比较次数 | 6 |  | 1 |  |  | 1 | 1 | 2 | 3 | 4 |

ASLsucc=18/7 ASLunsucc=32/10

26．（1）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 关键字 | 27 | 53 | 2 |  |  | 31 | 19 | 20 | 8 | 18 |  |  |  |
| 比较次数 | 3 | 1 | 1 |  |  | 1 | 1 | 1 | 1 | 2 |  |  |  |

（2）ASLsuss =11/8

27．（1）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 关键字 | 0 |  |  | 3 | 29 | 200 | 32 | 45 | 58 | 100 | 10 | 126 | 400 |
| 比较次数 | 1 |  |  | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 1 | 3 | 3 |

关键字29和45各发生一次碰撞，关键字58,126和400各发生两次碰撞，其余关键字无碰撞。

（2）

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 关键字 | 58 | 10 | 100 | 3 |  |  |  |  |  |  |  |  |  |