

一、填空题 ()

1. 在数据的存放无规律而言的线性表中进行检索的最佳方法是 顺序查找 (线性查找)。
2. 线性有序表 $(a_1, a_2, a_3, \dots, a_{256})$ 是从小到大排列的, 对一个给定的值 k , 用二分法检索表中与 k 相等的元素, 在查找不成功的情况下, 最多需要检索 8 次。设有 100 个结点, 用二分法查找时, 最大比较次数是 7。
3. 假设在有序线性表 $a[20]$ 上进行折半查找, 则比较一次查找成功的结点数为 1; 比较两次查找成功的结点数为 2; 比较四次查找成功的结点数为 8; 平均查找长度为 3.7。

解: 显然, 平均查找长度 = $O(\log_2 n) < 5$ 次 (2^5)。但具体是多少次, 则不应当按照公式

$ASL = \frac{n+1}{n} \log_2(n+1)$ 来计算 (即 $(21 \times \log_2 21) / 20 = 4.6$ 次并不正确!)。因为这是在假设 $n = 2^m - 1$ 的情况下

推导出来的公式。应当用穷举法罗列:

全部元素的查找次数为 $= (1 + 2 \times 2 + 4 \times 3 + 8 \times 4 + 5 \times 5) = 74$; $ASL = 74 / 20 = 3.7$!!!

4. 折半查找有序表 $(4, 6, 12, 20, 28, 38, 50, 70, 88, 100)$, 若查找表中元素 20, 它将依次与表中元素 28, 6, 12, 20 比较大小。
5. 在各种查找方法中, 平均查找长度与结点个数 n 无关的查找方法是 散列查找。
6. 散列法存储的基本思想是由 关键字的值 决定数据的存储地址。
7. 有一个表长为 m 的散列表, 初始状态为空, 现将 n ($n < m$) 个不同的关键码插入到散列表中, 解决冲突的方法是用线性探测法。如果这 n 个关键码的散列地址都相同, 则探测的总次数是 $n(n-1)/2 = (1+2+\dots+n-1)$ 。(而任一元素查找次数 $\leq n-1$)

二、单项选择题 ()

- (B) 1. 在表长为 n 的链表中进行线性查找, 它的平均查找长度为
A. $ASL = n$; B. $ASL = (n+1) / 2$;
C. $ASL = \sqrt{n} + 1$; D. $ASL \approx \log_2(n+1) - 1$
- (A) 2. 折半查找有序表 $(4, 6, 10, 12, 20, 30, 50, 70, 88, 100)$ 。若查找表中元素 58, 则它将依次与表中 比较大小, 查找结果是失败。
A. 20, 70, 30, 50 B. 30, 88, 70, 50 C. 20, 50 D. 30, 88, 50
- (C) 3. 对 22 个记录的有序表作折半查找, 当查找失败时, 至少需要比较 次关键字。
A. 3 B. 4 C. 5 D. 6
- (A) 4. 链表适用于 查找
A. 顺序 B. 二分法 C. 顺序, 也能二分法 D. 随机
- (C) 5. 折半搜索与二叉搜索树的时间性能
A. 相同 B. 完全不同 C. 有时不相同 D. 数量级都是 $O(\log_2 n)$

6. 从供选择的答案中, 选出应填入下面叙述___?___内的最确切的解答, 把相应编号写在答卷的对应栏内。

要进行线性查找, 则线性表___A___; 要进行二分查找, 则线性表___B___; 要进行散列查找, 则线性表___C___。某顺序存储的表格, 其中有 90000 个元素, 已按关键项的值的上升顺序排列。现假定对各个元素进行查找的概率是相同的, 并且各个元素的关键项的值皆不相同。当用顺序查找法查找时, 平均比较次数约为___D___, 最大比较次数为___E___。

供选择的答案:

- A~C: ① 必须以顺序方式存储 ② 必须以链表方式存储 ③ 必须以散列方式存储
④ 既可以以顺序方式, 也可以以链表方式存储
⑤ 必须以顺序方式存储且数据元素已按值递增或递减的次序排好
⑥ 必须以链表方式存储且数据元素已按值递增或递减的次序排好

D, E: ① 25000 ② 30000 ③ 45000 ④ 90000

答案: A=___④___ B=___⑤___ C=___③___ D=___③___ E=___④___

7. 从供选择的答案中, 选出应填入下面叙述___?___内的最确切的解答, 把相应编号写在答卷的对应栏内。

数据结构反映了数据元素之间的结构关系。链表是一种___A___, 它对于数据元素的插入和删除___B___。通常查找线性表数据元素的方法有___C___和___D___两种方法, 其中___C___是一种只适合于顺序存储结构但___E___的方法; 而___D___是一种对顺序和链式存储结构均适用的方法。

供选择的答案:

A: ① 顺序存储线性表 ② 非顺序存储非线性表 ③ 顺序存储非线性表 ④ 非顺序存储线性表

B: ① 不需要移动结点, 不需改变结点指针 ② 不需要移动结点, 只需改变结点指针
③ 只需移动结点, 不需改变结点指针 ④ 既需移动结点, 又需改变结点指针

C: ① 顺序查找 ② 循环查找 ③ 条件查找 ④ 二分法查找

D: ① 顺序查找 ② 随机查找 ③ 二分法查找 ④ 分块查找

E: ① 效率较低的线性查找 ② 效率较低的非线性查找

③ 效率较高的非线性查找 ④ 效率较高的线性查找

答案: A=___④___ B=___②___ C=___④___ D=___①___ E=___③___

8. 从供选择的答案中, 选出应填入下面叙述___?___内的最确切的解答, 把相应编号写在答卷的对应栏内。

在二叉排序树中, 每个结点的关键码值___A___, ___B___一棵二叉排序, 即可得到排序序列。同一个结点集合, 可用不同的二叉排序树表示, 人们把平均检索长度最短的二叉排序树称作最佳二叉排序, 最佳二叉排序树在结构上的特点是___C___。

供选择的答案

A: ① 比左子树所有结点的关键码值大, 比右子树所有结点的关键码值小

② 比左子树所有结点的关键码值小, 比右子树所有结点的关键码值大

③ 比左右子树的所有结点的关键码值都大

④ 与左子树所有结点的关键码值和右子树所有结点的关键码值无必然的大小关系

B: ① 前序遍历 ② 中序(对称)遍历 ③ 后序遍历 ④ 层次遍历

C: ① 除最下二层可以不满外, 其余都是充满的 ② 除最下一层可以不满外, 其余都是充满的

③ 每个结点的左右子树的高度之差的绝对值不大于 1 ④ 最下层的叶子必须在最左边

答案: A=___①___ B=___②___ C=___②___

9. 从供选择的答案中, 选出应填入下面叙述___?___内的最确切的解答, 把相应编号写在答卷的对应栏内。

散列法存储的基本思想是根据 A 来决定 B，碰撞（冲突）指的是 C，处理碰撞的两类主要方法是 D。

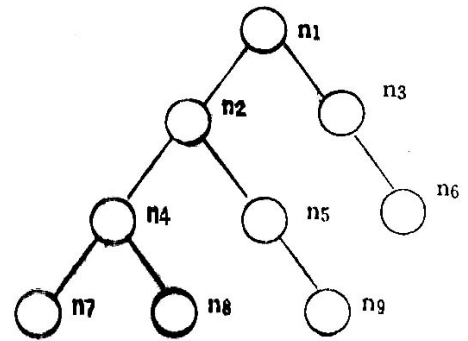
供选择的答案

- A, B: ① 存储地址 ② 元素的符号 ③ 元素个数 ④ 关键码值
 ⑤ 非码属性 ⑥ 平均检索长度 ⑦ 负载因子 ⑧ 散列表空间
- C: ① 两个元素具有相同序号 ② 两个元素的关键码值不同，而非码属性相同
 ③ 不同关键码值对应到相同的存储地址 ④ 负载因子过大 ⑤ 数据元素过多
- D: ① 线性探查法和双散列函数法 ② 建溢出区法和建不建溢出区法
 ③ 除余法和折叠法 ④ 拉链法和开地址法

答案: A = ④ B = ① C = ③ D = ④

10. 考虑具有如下性质的二叉树：除叶子结点外，每个结点的值都大于其左子树上的一切结点的值。并小于等于其右子树上的一切结点的值。

现把 9 个数 1, 2, 3, ..., 8, 9 填入右图所示的二叉树的 9 个结点中，并使之具有上述性质。此时，n1 的值是 A，n2 的值是 B，n9 的值是 C。现欲把 $\sqrt{10}$ 放入此树并使该树保持前述性质，增加的一个结点可以放在 D 或 E。

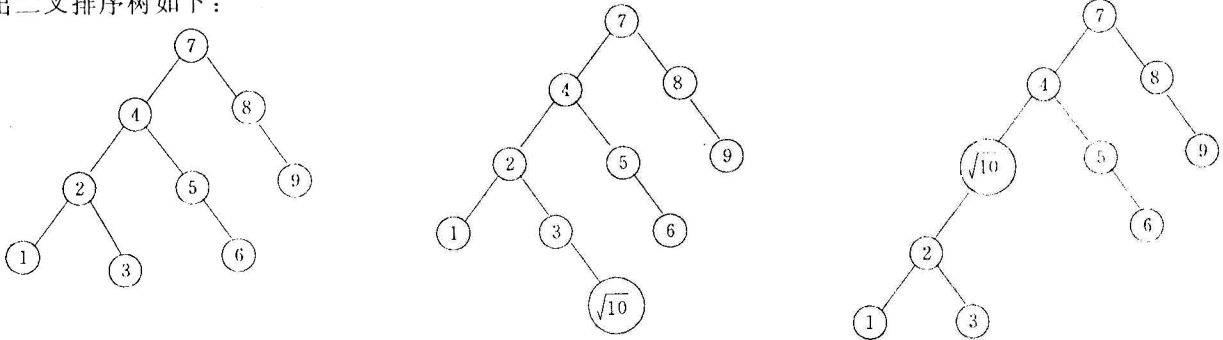


供选择的答案

- A~C: ① 1 ② 2 ③ 3 ④ 4 ⑤ 5
 ⑥ 6 ⑦ 7 ⑧ 8 ⑨ 9
- D~E: ① n7 下面 ② n8 下面 ③ n9 下面 ④ n6 下面
 ⑤ n1 与 n2 之间 ⑥ n2 与 n4 之间 ⑦ n6 与 n9 之间 ⑧ n3 与 n6 之间

答案: A = ⑦ B = ④ C = ⑥ D = ② E = ⑥

画出二叉排序树如下：



三、简答题（）

1. 对分（折半）查找适不适合链表结构的序列，为什么？用二分查找的查找速度必然比线性查找的速度快，这种说法对吗？

答：不适合！虽然有序的单链表的结点是按从小到大（或从大到小）顺序排列，但因其存储结构为单链表，查找结点时只能从头指针开始逐步搜索，故不能进行折半查找。

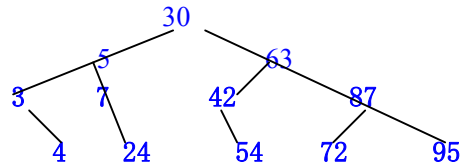
二分查找的速度在一般情况下是快些，但在特殊情况下未必快。例如所查数据位于首位时，则线性查找快；而二分查找则慢得多。

2.假定对有序表：(3, 4, 5, 7, 24, 30, 42, 54, 63, 72, 87, 95) 进行折半查找，试回答下列问题：

- (1) 画出描述折半查找过程的判定树；
- (2) 若查找元素 54，需依次与哪些元素比较？
- (3) 若查找元素 90，需依次与哪些元素比较？
- (4) 假定每个元素的查找概率相等，求查找成功时的平均查找长度。

解：

(1) 先画出判定树如下（注： $mid = \lfloor (1+12)/2 \rfloor = 6$ ）：



- (2) 查找元素 54，需依次与 30, 63, 42, 54 等元素比较；
- (3) 查找元素 90，需依次与 30, 63, 87, 95, 72 等元素比较；
- (4) 求 ASL 之前，需要统计每个元素的查找次数。判定树的前 3 层共查找 $1+2 \times 2+4 \times 3=17$ 次；但最后一层未滿，不能用 8×4 ，只能用 $5 \times 4=20$ 次，所以 $ASL = 1/12 (17+20) = 37/12 \approx 3.08$

3. 用比较两个元素大小的方法在一个给定的序列中查找某个元素的时间复杂度下限是什么？如果要求时间复杂度更小，你采用什么方法？此方法的时间复杂度是多少？

答：查找某个元素的时间复杂度下限，如果理解为最短查找时间，则当关键字值与表头元素相同时，比较 1 次即可。要想降低时间复杂度，可以改用 Hash 查找法。此方法对表内每个元素的比较次数都是 0 (1)。

4. 设哈希 (Hash) 表的地址范围为 0~17，哈希函数为： $H(K) = K \text{ MOD } 16$ 。

K 为关键字，用线性探测法再散列法处理冲突，输入关键字序列：

(10, 24, 32, 17, 31, 30, 46, 47, 40, 63, 49)

造出 Hash 表，试回答下列问题：

- (1) 画出哈希表的示意图；
- (2) 若查找关键字 63，需要依次与哪些关键字进行比较？
- (3) 若查找关键字 60，需要依次与哪些关键字比较？
- (4) 假定每个关键字的查找概率相等，求查找成功时的平均查找长度。

解：(1) 画表如下：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
32	17	63	49					24	40	10				30	31	46	47

(2) 查找 63, 首先要与 $H(63)=63\%16=15$ 号单元内容比较，即 63 vs 31, no;

然后顺移，与 46, 47, 32, 17, 63 相比，一共比较了 6 次！

(3) 查找 60, 首先要与 $H(60)=60\%16=12$ 号单元内容比较，但因为 12 号单元为空 (应当有空标记)，所以应当只比较这一次即可。

(4) 对于黑色数据元素，各比较 1 次；共 6 次；

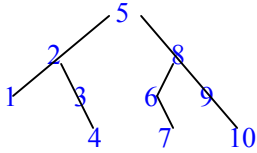
对红色元素则各不相同，要统计移位的位数。“63”需要 6 次，“49”需要 3 次，“40”需要 2 次，“46”需要 3 次，“47”需要 3 次，

所以 $ASL = 1/11 (6+2+3 \times 3) = 17/11 = 1.5454545454 \approx 1.55$

四、分析题（）

1. 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。

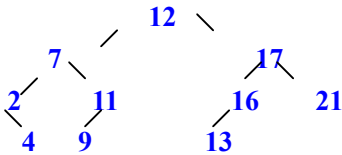
解：判定树应当描述每次查找的位置：



$$\begin{aligned}
 ASL &= 1/10 (1 + 2 \times 2 + 3 \times 4 + 4 \times 3) \\
 &= 1/10 (1 + 4 + 12 + 12) \\
 &= 29/10 = 2.9
 \end{aligned}$$

2. 在一棵空的二叉查找树中依次插入关键字序列为 12, 7, 17, 11, 16, 2, 13, 9, 21, 4, 请画出所得到的二叉查找树。

答：



验算方法： 用中序遍历应得到排序结果： 2,4,7,9,11,12,13,16,17, 21

3. 已知如下所示长度为 12 的表：

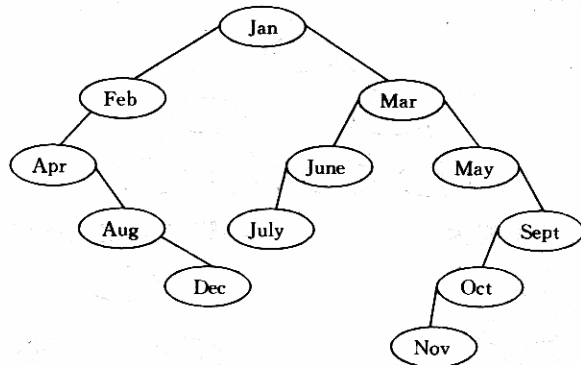
(Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec)

- 试按表中元素的顺序依次插入一棵初始为空的二叉排序树，画出插入完成之后的二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。
- 若对表中元素先进行排序构成有序表，求在等概率的情况下对此有序表进行折半查找时查找成功的平均查找长度。
- 按表中元素顺序构造一棵平衡二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。

解：

(1) 求得的二叉排序树如下图所示，在等概率情况下查找成功的平均查找长度为

$$ASL_{succ} = \frac{1}{12} (1 \times 1 + 2 \times 2 + 3 \times 3 + 4 \times 3 + 5 \times 2 + 6 \times 1) = \frac{42}{12}$$



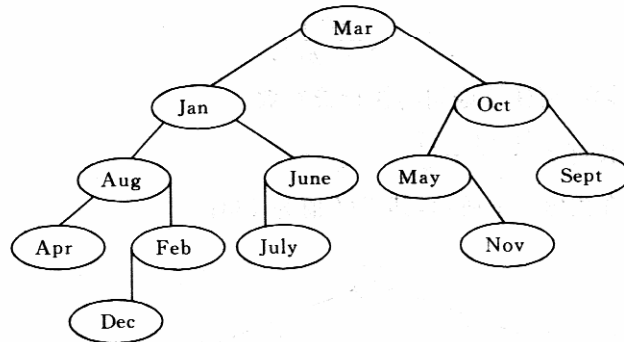
(2) 经排序后的表及在折半查找时找到表中元素所经比较的次数对照如下:

Apr	Aug	Dec	Feb	Jan	July	June	Mar	May	Nov	Oct	Sept
3	4	2	3	4	1	3	4	2	4	3	4

等概率情况下查找成功时的平均查找长度为

$$ASL_{succ} = \frac{1}{12}(1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 5) = \frac{37}{12}$$

(3) 按教科书 9.2.1 节所述求得的平衡二叉树为



它在等概率情况下的平均查找长度为

$$ASL = \frac{1}{12}(1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 4 + 5 \times 1) = \frac{38}{12}$$

4. 选取散列函数 $H(\text{key}) = (3 * \text{key}) \% 11$, 用线性探测法处理冲突, 对下列关键码序列构造一个散列地址空间为 $0 \sim 10$, 表长为 11 的散列表, $\{22, 41, 53, 08, 46, 30, 01, 31, 66\}$ 。

解: 由题意知, $m=11$ (刚好为素数)

地址	值	链接指针
0	22	1
1	66	
2	41	3
3	08	4,7
4	30	
5	53	
6	46	
7	01	
8	31	

则 $(22*3)\%11=6\cdots\cdots 0$
 $(41*3)\%11=11\cdots\cdots 2$
 $(53*3)\%11=14\cdots\cdots 5$
 $(08*3)\%11=2\cdots\cdots 2$
 $(46*3)\%11=12\cdots\cdots 6$
 $(30*3)\%11=8\cdots\cdots 2$
 $(01*3)\%11=0\cdots\cdots 3$
 $(31*3)\%11=8\cdots\cdots 5$
 $(66*3)\%11=9\cdots\cdots 0$

22	66	41	8	30	53	46	1	31		
0	1	2	3	4	5	6	7	8	9	10
1		3	4, 7							

五、算法设计题 ()

1. 已知 11 个元素的有序表为 (05 13 19 21 37 56 64 75 80 88 92), 请写出折半查找的算法程序, 查找关键字为 key 的数据元素 (建议上机调试)。

解：折半查找的 C 程序有很多参考资料，注意此题要求是整型量。

折半查找的一个递归算法如下，形式非常简洁！

```
int Search_Bin_Recursive(SSTable ST, int key, int low, int high) //折半查找的递归算法
{
    if(low>high) return 0; //查找不到时返回 0
    mid=(low+high)/2;
    if(ST.elem[mid].key==key) return mid;
    else if(ST.elem[mid].key>key)
        return Search_Bin_Recursive(ST, key, low, mid-1);
    else return Search_Bin_Recursive(ST, key, mid+1, high);
}
//Search_Bin_Recursive
```

2. 试写一个判别给定二叉树是否为二叉排序树的算法，设此二叉树以二叉链表作存储结构。且树中结点的关键字均不同。

解：注意仔细研究二叉排序树的定义。易犯的典型错误是按下述思路进行判别：“若一棵非空的二叉树其左、右子树均为二叉排序树，且左子树的根的值小于根结点的值，又根结点的值不大于右子树的根的值，则是二叉排序树”

（刘注：即不能只判断左右孩子的情况，还要判断左右孩子与双亲甚至根结点的比值也要遵循（左小右大）原则）。

若要采用递归算法，建议您采用如下的函数首部：

bool BisortTree(BiTree T, BiTree&PRE)，其中 PRE 为指向当前访问结点的前驱的指针。

（或者直接存储前驱的数值，随时与当前根结点比较）

一个漂亮的算法设计如下：

```
int last=0, flag=1; // last 是全局变量，用来记录前驱结点值，只要每个结点都比前驱大就行。
int Is_BSTree(Bitree T) //判断二叉树 T 是否二叉排序树，是则返回 1，否则返回 0
{
    if(T->lchild&&flag) Is_BSTree(T->lchild);
    if(T->data<last) flag=0; //与其中序前驱相比较, flag=0 表示当前结点比直接前驱小，则立即返回
    last=T->data;
    if(T->rchild&&flag) Is_BSTree(T->rchild);
    return flag;
}
//Is_BSTree
```

3. 已知一个含有 1000 个记录的表，关键字为中国人姓氏的拼音，请给出此表的一个哈希表设计方案，要求它在等概率情况下查找成功的平均查找长度不超过 3。

解：设计哈希表的步骤为：

- 根据所选择的处理冲突的方法求出装载因子 a 的上界；
- 由 a 值设计哈希表的长度 m ；
- 根据关键字的特性和表长 m 选定合适的哈希函数。

刘注：要求 $ASL \leq 3$ ，则 m 必然要尽量长，以减少冲突；

4. 已知某哈希表的装载因子小于 1，哈希函数 $H(\text{key})$ 为关键字（标识符）的第一个字母在字母表中的序号，处理冲突的方法为线性探测开放定址法。试编写一个按第一个字母的顺序输出哈希表中所有关键字的算法。

解：注意此题给出的条件：装载因子 $\alpha < 1$ ，则哈希表未填满。由此可写出下列形式简明的算法：

```
void PrintWord(Hash Table ht)
```

```
{//按第一个字母的顺序输出哈希表 ht 中的标识符。哈希函数为表示符的第一个字母在字母表中的序号，  
处理冲突的方法是线性探测开放定址法。
```

```
for(i=1; i<=26; i++){
```

```
  j=i;
```

```
  While(ht.elem[j].key){
```

```
    if(Hash(ht.elem[j].key==i)printf(ht.elem[j].key);
```

```
    j=(j+1)%m;
```

```
  }
```

```
}
```

```
//PrintWord
```