计算机软件技术基础题库与上机实践

夏清国 编

西北工业大学出版社

计算机软件技术基础题库与上机实践

夏清国　编

西北工业大学出版社

【内容简介】 本书是高等学校非计算机专业开设的“计算机软件技术基础”课程的配套用书，全书分为习题与解答、上机实践两个部分。习题与解答部分包括各种题型的题目（如选择题、填空题、名词解释、简答题、应用题等）及答案。实践部分共有12个实验，涵盖了数据结构、面向对象的软件开发技术、操作系统、数据库管理系统、软件工程等内容，每个实验都给出实验提示，以帮助学生顺利完成实验和加深对概念的理解。本书适用于在校学习“计算机软件技术基础”课程的学生以及相关自学人员，也可作为教师参考书。

图书在版编目(CIP)数据

计算机软件技术基础题库与上机实践/夏清国编．—西安：西北工业大学出版社，2003.9

ISBN 7-5612-1689-0

Ⅰ.计… Ⅱ.夏… Ⅲ.软件—高等学校—教材参考资料 Ⅳ.TP31

中国版本图书馆CIP数据核字(2003)第070731号

出版发行：西北工业大学出版社
通信地址：西安市友谊西路127号 邮编：710072 电话：(029)8493844
网　　址：www.nwpup.com
印 刷 者：西安东江印务有限公司
开　　本：787 mm×960 mm 1/16
印　　张：12
字　　数：227千字
版　　次：2003年9月第1版 2003年9月第1次印刷
印　　数：1～6 000册
定　　价：16.00元

前　言

计算机软件技术基础是在学生了解计算机基础知识的基础上，为提高学生对软件本质的理解，通过对计算机软件基础领域的基本原理、方法和思想的学习，提高对软件开发工具和环境的适应能力的一门必修课程。为了帮助高校学生及自学人员学好该课程，顺利地通过考试，我们在积累多年教学经验的基础上，编写了此书。本书是高等学校非计算机专业开设的“计算机软件技术基础”课程的配套用书，全书为习题与解答、上机实践两个部分。习题与解答部分包括各种题型的题目（选择题、填空题、名词解释、简答题、应用题等）及相应的参考答案；实践部分共有 12 个实验，涵盖了数据结构、面向对象的软件开发技术、操作系统、数据库管理系统、软件工程等内容，每个实验都给出了实验提示，以帮助学生顺利完成实验和加深对概念的理解。限于作者水平有限，书中不足与疏漏之处在所难免，恳请各位专家和广大读者批评指正。

作　者

2003 年 7 月

目　录

第一部分　习题与解答

第二部分　上机实践

第一部分　习题与解答

第1章　数据结构

一、选择题

1. 算法指的是(　　)。

A. 计算机程序　　B. 解决问题的计算方法

C. 排序方法　　D. 解决问题的有限运算序列

2. 在数据的树形结构中，数据元素之间为(　　)的关系。

A. 0∶0　　B. 1∶1　　C. 1∶n　　D. m∶n

3. 数据的存储结构包括顺序、链接、散列和(　　)4种基本类型。

A. 索引　　B. 数组　　C. 集合　　D. 向量

4. 一个数组元素 $a[i]$ 与(　　)的表示等价。

A. $\&a+i$　　B. $*(a+i)$　　C. $*a+i$　　D. $a+i$

5. 若只需要利用形参间接访问实参指针所指向的对象，而形参本身具有相应的存储空间，则应把形参变量说明为(　　)参数。

A. 指针　　B. 引用　　C. 值　　D. 指针引用

6. 若只需要利用形参实现对实参值的拷贝，函数体操作形参时与实参无关，则应把形参变量说明为(　　)参数。

A. 指针　　B. 引用　　C. 值　　D. 指针引用

7. 下面程序段的时间复杂性的量级为(　　)。

```
int i =0,s1 =,s2 =0;
while(i ++ <n)
  { if(i%2) s1 +=i;
    else s2 +=i;
  }
```

A. $O(1)$　　B. $O(\mathrm{lb}n)$　　C. $O(n)$　　D. $O(2n)$

8. 下面程序段的时间复杂度为(　　)。

```
for (int i=0; i<m; i++)
```

```
for (int j =0; j <n; j ++)
  a[i][j] =i * j;
```

A. $O(m^2)$　　B. $O(n^2)$　　C. $O(m+n)$　　D. $O(m*n)$

9. 执行下面程序段时，S 语句的执行次数为(　　)。

```
for(int i=1; i<=n; i++)
  for(int j=1; j<=i; j++)
    S;
```

A. $n(n-1)/2$　　B. $n(n+1)/2$　　C. $n^2/2$　　D. n

10. 在一个长度为 n 的顺序存储的线性表中，向第 i 个元素($1 \leqslant i \leqslant n+1$)位置插入一 个新元素时，需要从后向前依次后移(　　)个元素。

A. $n-i$　　B. $n-i+1$　　C. $n-i-1$　　D. i

11. 在一个长度为 n 的顺序存储的线性表中，删除第 i 个元素($1 \leqslant i \leqslant n$)时，需要从前向后依次前移(　　)个元素。

A. $n-i$　　B. $n-i+1$　　C. $n-i-1$　　D. i

12. 在一个长度为 n 的线性表中，删除值为 x 的元素时需要比较元素和移动元素的总次数为(　　)。

A. $(n+1)/2$　　B. $n/2$　　C. n　　D. $n+1$

13. 在一个顺序表的表尾插入一个元素的时间复杂度为(　　)。

A. $O(n)$　　B. $O(1)$　　C. $O(n*n)$　　D. $O(\mathrm{lb}n)$

14. 在一个顺序表中任何位置插入一个元素的时间复杂度为(　　)。

A. $O(n)$　　B. $O(n/2)$　　C. $O(1)$　　D. $O(n^2)$

15. 在一个单链表中删除 p 所指向结点的后继结点时，其算法的时间复杂度为(　　)。

A. $O(n)$　　B. $O(n/2)$　　C. $O(1)$　　D. $O(n^2)$

16. 线性表的链式存储比顺序存储更有利于进行(　　)操作。

A. 查找　　B. 表尾插入或删除

C. 按值插入或删除　　D. 表头插入或删除

17. 线性表的顺序存储比链式存储更有利于进行(　　)操作。

A. 查找　　B. 表尾插入或删除

C. 按值插入或删除　　D. 表头插入或删除

18. 在一个单链表中，若要在 p 所指向的结点之后插入一个新结点，则需要相继修改(　　)个指针域的值。

A. 1　　B. 2　　C. 3　　D. 4

19. 在一个带头结点的循环双向链表中，若要在 p 所指向的结点之前插入一个新结 点，则需要相继修改(　　)个指针域的值。

A. 2　　B. 3　　C. 4　　D. 6

20. 在一个表头指针为 ph 的单链表中，若要向表头插入一个由指针 p 指向的结点，则应执行(　　)操作。

A. ph=p;p->next=ph;

B. p->next=ph;ph=p;

C. p->next=ph;p=ph;

D. p->next=ph->next;ph->next=p;

21. 在一个表头指针为 ph 的单链表中，若要在指针 q 所指结点的后面插入一个由指针 p 所指向的结点，则执行(　　)操作。

A. q->next=p->next;p->next=q;

B. p->next=q->next;q=p;

C. q->next=p->next;p->next=q;

D. p->next=q->next;q->next=p

22. 在一个单链表 HL 中，若要删除由指针 q 所指向结点的后继结点(若存在的话)，则执行(　　)操作。

A. p=q->next;p->next=q->next;

B. p=q->next;q->next=p;

C. p=q->next;q->next=p->next;

D. q->next=q->next->next;q->next=q;

23. 在一个带头结点的循环双向链表中，若要在指针 p 所指向的结点之后插入一个 q 指针所指向的结点，则需要对 q->next 赋值为(　　)。

A. P->prior　　B. p->next

C. p->next->next　　D. p->prior->prior

24. 在一个带头结点的循环双向链表中，若要在指针 p 所指向的结点之前插入一个 q 指针所指向的结点，则需要对 p->prior->next 赋值为(　　)。

A. q　　B. p　　C. p->next　　D. p->prior

25. 在一个带头结点的循环双向链表中，若要删除指针 p 所指向的结点则执行(　　)操作。

A. p->prior->next=
p->next;p->next->prior=p->prior;

B. p->next->prior=p;p->next=p->next->next;

C. p->prior->next=p;p->next=p->next->prior;

D. p=p->next;p->prior->next=p->prior;

26. 栈的插入和删除操作在(　　)进行。

A. 栈顶　　B. 栈底　　C. 任意位置　　D. 指定位置

27. 当利用大小为 N 的数组顺序存储一个栈时，假定用 top==N 表示栈空，则向这个 栈插入一个元素时，首先应执行(　　)语句修改 top 指针。

A. top++;　　B. top--;　　C. top=0;　　D. top=N-1;

28. 假定利用数组 $a[N]$顺序存储一个栈，用 top 表示栈顶指针，用 top==N+1 表示栈 空，该数组所能存储的栈的最大长度为 N，则表示栈满的条件为(　　)。

A. top==1;　　B. top==-1;　　C. top==0;　　D. top>1;

29. 假定利用数组 $a[N]$顺序存储一个栈，用 top 表示栈顶指针，top==-1 表示栈空，并已知栈未满，当元素 x 进栈时所执行的操作为(　　)。

A. a[--top]=x;　　B. a[top--]=x;

C. a[++top]=x;　　D. a[top++]= x;

30. 假定利用数组 $a[N]$顺序存储一个栈，用 top 表示栈顶指针，top==-1 表示栈空，并已知栈未空，当退栈并返回栈顶元素时所执行的操作为(　　)。

A. return a[--top];　　B. return a[top--];

C. return a[++top];　　D. return a[top++];

31. 假定一个链式栈的栈顶指针用 top 表示，该链式栈为空的条件为(　　)。

A. top! =NULL;　　B. top==top->next;

C. top==NULL;　　D. top! =top->next;

32. 假定一个链式栈的栈顶指针用 top 表示，每个结点的结构为 data|next，当 p 所指向的结点进栈时，执行的操作为(　　)。

A. p->next=top;top=top->next;

B. top=p;p->next=top;

C. p->next=top->next;top->next=p;

D. p->next=top;top=p;

33. 假定一个链式栈的栈顶指针用 top 表示，每个结点的结构为 data|next，退栈时所执行的指针操作为(　　)。

A. top->next=top;

B. top=top->data;

C. top=top->next;

D. top->next=top->next->next;

34. 若让元素 1,2,3,4 依次进栈，则出栈次序不可能出现(　　)的情况。

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

35. 在一个顺序循环队列中，队首指针指向队首元素的(　　)位置。

A. 前一个　　B. 后一个　　C. 当前　　D. 最后

36. 当利用大小为 N 的数组循环顺序存储一个队列时，该队列的最大长

度为(　　)。

A. $N-2$　　B. $N-1$　　C. N　　D. $N+1$

37. 从一个顺序循环队列中删除元素时,首先需要(　　)。

A. 前移队首指针

B. 后移队首指针

C. 取出队首指针所指位置上的元素

D. 取出队尾指针所指位置上的元素

38. 假定一个顺序循环队列的队首和队尾指针分别用 f 和 r 表示,则判断队空的条件为(　　)。

A. f+1==r　　B. r+1==f　　C. f==0　　D. f==r

39. 假定一个顺序循环队列存储于数组 $a[N]$中,其队首和队尾指针分别用 f 和 r 表示,则判断队满的条件为(　　)。

A. (r-1)%N==f　　B. (r+1)%N==f

C. (f-1)%N==r　　D. (f+1)%N==r

40. 假定利用数组 $a[N]$循环顺序存储一个队列,用 f 和 r 分别表示队首和队尾指针,并已知队列未满,当元素 x 入列时所执行的操作为(　　)。

A. a[++r%N]=x;　　B. a[r++%N]=x;

C. a[--r%N]=x;　　D. a[r--%N]=x;

41. 假定利用数组 $a[N]$循环顺序存储一个队列,用 f 和 r 分别表示队首和队尾指针,并已知队未空,当出列并返回队首元素时所执行的操作为(　　)。

A. return a[++r%N];　　B. return a[--r%N];

C. return a[++f%N];　　D. return a[f++%N];

42. 假定一个链式队列的队首和队尾指针分别为 front 和 rear,则判断队空的条件为(　　)。

A. front==rear　　B. front! =NULL

C. rear! =NULL　　D. front==NULL

43. 假定一个链式队列的队首和队尾指针分别用 front 和 rear 表示,每个结点的结构为:data|next,当出列时所进行的指针操作为(　　)。

A. front=front->next;

B. rear=rear->next;

C. front->next=rear;rear=rear->next;

D. front=front->next;front+next=rear;

44. 假定一个带头结点的循环链式队列的队首和队尾指针分别用 front 和 rear 表示,则判断队空的条件为(　　)。

A. front==rear->next　　B. rear==NULL

C. front==NULL　　D. front==rear

45. 假定一个带头结点的循环链式队列的队首和队尾指针分别用 front 和 rear 表示，每个结点包含值域 data 和指针域 next，则使 p 所指结点入列所执行的操作为(　　)。

A. p->next=NULL;rear=rear->next=p;

B. p->next=rear->next;rear=rear->next=p;

C. p->next=front;front=p;

D. p->next=front->next;front->next=p;

46. 在一个长度为 N 的数组空间中，循环顺序存储着一个队列，该队列的队首和队尾指针分别用 front 和 rear 表示，则该队列中的元素个数为(　　)。

A. (rear-front)%N　　B. (rear-front+N)%N

C. (rear+N)%N　　D. (front+N)%N

47. 二维数组 $A[12,10]$采用行优先存储，每个数据元素占用 4 个存储单元，该数组的首地址(即 $A[0,0]$的地址)为 1 200，则 $A[6,5]$的地址为(　　)。

A. 1 400　　B. 1 404　　C. 1 372　　D. 1 460

48. 有一个 $M\times N$ 的矩阵 $\boldsymbol{A}$，若采用行优先进行顺序存储，每个元素占用 8 个字节，则 $A_{ij}(1\leqslant i\leqslant M,1\leqslant j\leqslant N)$元素的相对字节地址(相对首元素地址而言)为(　　)。

A. $((i-1)\times N+j)\times 8$　　B. $((i-1)\times N+j-1)\times 8$

C. $(i\times N+j-1)\times 8$　　D. $((i-1)\times N+j+1)\times 8$

49. 有一个 $N\times N$ 的下三角矩阵 $\boldsymbol{A}$，若采用行优先进行顺序存储，每个元素占用 k 个字节，则 $A_{ij}(1\leqslant i\leqslant N,1\leqslant j\leqslant i)$元素的相对字节地址(相对首元素地址而言)为(　　)。

A. $(i\times(i+1)/2+j-1)\times 4$　　B. $(i\times i/2+j)\times 4$

C. $(i\times(i-1)/2+j-1)\times 4$　　D. $(i\times(i-1)/2+j)\times 4$

50. 树中所有结点的度等于所有结点数加(　　)。

A. 0　　B. 1　　C. -1　　D. 2

51. 在一棵树中，(　　)没有前趋结点。

A. 树枝结点　　B. 叶子结点　　C. 树根结点　　D. 空结点

52. 在一棵树中，每个结点最多有(　　)个前趋结点。

A. 0　　B. 1　　C. 2　　D. 任意多个

53. 在一棵二叉树的二叉链表中，空指针域数等于非空指针域数加(　　)。

A. 2　　B. 1　　C. 0　　D. -1

54. 在一棵具有 n 个结点的二叉树中，所有结点的空子树个数等于(　　)。

A. n　　B. $n-1$　　C. $n+1$　　D. $2n$

55. 在一棵具有 n 个结点的二叉树的第 i 层上，最多具有(　　)个结点。

A. 2^i　　B. 2^{i-1}　　C. 2^{i-1}　　D. 2^n

56. 在一棵深度为 h 的完全二叉树中，所含结点个数不小于(　　)。

A. 2^h　　B. 2^{h+1}　　C. 2^{h-1}　　D. 2^{h-1}

57. 在一棵深度为 h 的完全二叉树中，所含结点个数不大于(　　)。

A. 2^h　　B. 2^{h-1}　　C. 2^{h-1}　　D. 2^{h-1}

58. 在一棵具有 35 个结点的完全二叉树中，该树的深度为(　　)。

A. 6　　B. 7　　C. 5　　D. 8

59. 在一棵完全二叉树中，若编号为 i 的结点存在左孩子，则左孩子结点的编号为(　　)。

A. $2i$　　B. $2i-1$　　C. $2i+1$　　D. $2i+2$

60. 在一棵完全二叉树中，若编号为 i 的结点存在右孩子，则右孩子结点的编号为(　　)。

A. $2i$　　B. $2i-1$　　C. $2i+1$　　D. $2i+2$

61. 在一棵完全二叉树中，对于编号为 $i(i>1)$ 的结点，其双亲结点的编号为(　　)。

A. $(i+1)/2$　　B. $(i-1)/2$　　C. $i\%2$　　D. $i/2$

62. 有如图 1.1 所示的一棵二叉树，则该二叉树所含的单支结点数为(　　)。

A. 2　　B. 3　　C. 4　　D. 5

63. 有如图 1.2 所示的一棵二叉树，则该二叉树的中序遍历序列为(　　)。

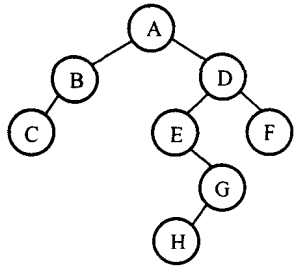

图　1.1

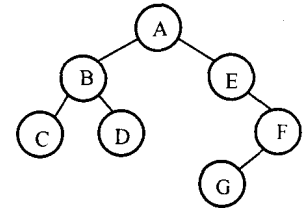

图　1.2

A. A,B,C,D,E,F,G　　B. C,D,B,G,F,E,A

C. C,B,D,A,E,G,F　　D. A,B,E,C,D,F,G

64. 有如图 1.2 所示的一棵二叉树，则该二叉树的先序遍历序列为(　　)。

A. A,B,C,D,E,F,G　　B. C,D,B,G,F,E,A

C. C,B,D,A,E,G,F　　D. A,B,E,C,D,F,G

65. 有如图 1.2 所示的一棵二叉树，则该二叉树的后序遍历序列为(　　)。

A. A,B,C,D,E,F,G　　B. C,D,B,G,F,E,A

C. C,B,D,A,E,G,F　　D. A,B,E,C,D,F,G

66. 利用 n 个值生成的哈夫曼树中共有(　　)个结点。

A. n　　B. $n+1$　　C. $2n$　　D. $2n-1$

67. 利用 3,6,8,12 这 4 个值作为叶子结点的权,生成一棵哈夫曼树,该树的带权路径长度为(　　)。

A. 55　　B. 29　　C. 58　　D. 38

68. 在一个具有 n 个顶点的有向图中,若所有顶点的出度数之和为 s,则所有顶点的入度数之和为(　　)。

A. s　　B. $s-1$　　C. $s+1$　　D. n

69. 在一个具有 n 个顶点的有向图中,若所有顶点的出度数之和为 s,则所有顶点的度数之和为(　　)。

A. s　　B. $s-1$　　C. $s+1$　　D. $2s$

70. 在一个具有 n 个顶点的无向图中,若具有 e 条边,则所有顶点的度数之和为(　　)。

A. n　　B. e　　C. $n+e$　　D. $2e$

71. 在一个具有 n 个顶点的无向完全图中,所含的边数为(　　)。

A. n　　B. $n(n-1)$　　C. $n(n-1)/2$　　D. $n(n+1)/2$

72. 在一个具有 n 个顶点的有向完全图中,所含的边数为(　　)。

A. n　　B. $n(n-1)$　　C. $n(n-1)/2$　　D. $n(n+1)/2$

73. 对于一个具有 n 个顶点的无向连通图,它包含的连通分量的个数为(　　)。

A. 0　　B. 1　　C. n　　D. $n+1$

74. 若一个图中包含有 k 个连通分量,若按照深度优先搜索的方法访问所有顶点,则必须调用(　　)次深度优先搜索遍历的算法。

A. k　　B. 1　　C. $k-1$　　D. $k+1$

75. 若要把 n 个顶点连接为一个连通图,则至少需要(　　)条边。

A. n　　B. $n+1$　　C. $n-1$　　D. $2n$

76. 在一个具有 n 个顶点和 e 条边的无向图的邻接矩阵中,表示边存在的元素(又称 为有效元素)的个数为(　　)。

A. n　　B. ne　　C. e　　D. $2e$

77. 在一个具有 n 个顶点和 e 条边的有向图的邻接矩阵中,表示边存在的元素个数为(　　)。

A. n　　B. ne　　C. e　　D. $2e$

78. 在一个具有 n 个顶点和 e 条边的无向图的邻接表中,边结点的个数

为(　　)

A. n　　B. ne　　C. e　　D. $2e$

79. 对于一个有向图，若一个顶点的度为 $k1$，出度为 $k2$，则对应邻接表中该顶点单链表中的边结点数为(　　)。

A. $k1$　　B. $k2$　　C. $k1-k2$　　D. $k1+k2$

80. 对于一个有向图，若一个顶点的度为 $k1$，出度为 $k2$，则对应逆邻接表中该顶点单链表中的边结点数为(　　)。

A. $k1$　　B. $k2$　　C. $k1-k2$　　D. $k1+k2$

81. 对于一个无向图，下面(　　)的说法是正确的。

A. 每个顶点的入度等于出度

B. 每个顶点的度等于其入度与出度之和

C. 每个顶点的入度为 0

D. 每个顶点的出度为 0

82. 在一个有向图的邻接表中，每个顶点单链表中结点的个数等于该顶点的(　　)。

A. 出边数　　B. 入边数　　C. 度数　　D. 度数减 1

83. 若一个图的边集为{(A,B),(A,C),(B,D),(C,F),(D,E),(D,F)}，则从顶点 A 开始对该图进行深度优先搜索，得到的顶点序列可能为(　　)。

A. A,B,C,F,D,E　　B. A,C,F,D,E,B

C. A,B,D,C,F,E　　D. A,B,D,F,E,C

84. 若一个图的边集为{(A,B),(A,C),(B,D),(C,F),(D,E),(D,F)}，则从顶点 A 开始对该图进行广度优先搜索，得到的顶点序列可能为(　　)。

A. A,B,C,D,E,F　　B. A,B,C,F,D,E

C. A,B,D,C,E,F　　D. A,C,B,F,D,E

85. 若一个图的边集为{<1,2>,<1,4>,<2,5>,<3,1>,<3,5>,<4,3>}，则从顶点 1 开始对该图进行深度优先搜索，得到的顶点序列可能为(　　)。

A. 1,2,5,4,3　　B. 1,2,3,4,5

C. 1,2,5,3,4　　D. 1,4,3,2,5

86. 若一个图的边集为{<1,2>,<1,4>,<2,5>,<3,1>,<3,5>,<4,3>}，则从顶点 1 开始对该图进行广度优先搜索，得到的顶点序列可能为(　　)。

A. 1,2,3,4,5　　B. 1,2,4,3,5

C. 1,2,4,5,3　　D. 1,4,2,5,3

87. 由一个具有 n 个顶点的连通图生成的最小生成树中有(　　)条边。

A. n　　B. $n-1$　　C. $n+1$　　D. $2n$

88. 若查找每个元素的概率相等，则在长度为 n 的顺序表上查找任一元素的平均查找长度为(　　)。

A. n　　B. $n+1$　　C. $(n-1)/2$　　D. $(n+1)/2$

89. 对长度为 n 的单链有序表，若查找每个元素的概率相等，则查找任一元素的平均查找长度为(　　)。

A. $n/2$　　B. $(n+1)/2$　　C. $(n-1)/2$　　D. $n/4$

90. 对于长度为 9 的顺序存储的有序表，若采用二分查找，在等概率情况下的平均查找长度为(　　)的值除以 9。

A. 20　　B. 18　　C. 25　　D. 22

91. 对于长度为 18 的顺序存储的有序表，若采用二分查找，则查找第 15 个元素的查找长度为(　　)。

A. 3　　B. 4　　C. 5　　D. 6

92. 对于顺序存储的有序表(5,12,20,26,37,42,46,50,64)，若采用二分查找，则查找元素 26 的查找长度为(　　)。

A. 2　　B. 3　　C. 4　　D. 5

93. 在分块查找中，若用于保存数据元素的主表长度为 n，它被均分为 k 个子表，每个子表的长度均为 n/k，若用顺序查找确定块，则分块查找的平均查找长度为(　　)。

A. $n+k$　　B. $k+n/k$　　C. $(k+n/k)/2$　　D. $(k+n/k)/2+1$

94. 在分块查找中，若用于保存数据元素的主表长度为 144，它被均分为 12 个子表，每个子表的长度均为 12，若用顺序查找确定块，则分块查找的平均查找长度为(　　)。

A. 13　　B. 24　　C. 12　　D. 79

95. 在一棵深度为 h 的具有 n 个元素的二叉排序树中，查找所有元素的最长查找长度为(　　)。

A. n　　B. $\mathrm{lb}n$　　C. $(h+1)/2$　　D. h

96. 在一棵平衡二叉排序树中，每个结点的平衡因子的取值范围是(　　)。

A. $-1\sim1$　　B. $-2\sim2$　　C. $1\sim2$　　D. $0\sim1$

97. 若根据查找表(23,44,36,48,52,73,64,58)建立线性哈希表，采用 $H(K)=K\%13$ 计算哈希地址，则元素 64 的哈希地址为(　　)。

A. 4　　B. 8　　C. 12　　D. 13

98. 若根据查找表(23,44,36,48,52,73,64,58)建立线性哈希表，采用 $H(K)=K\%7$ 计算哈希地址，则哈希地址等于 3 的元素个数(　　)。

A. 1　　B. 2　　C. 3　　D. 4