上 海 交 通 大 学 试 卷（C 卷）

（ 2019 至 2020 学年 第 2 学期 ）

班级号 _____ 学号_____ 姓名 _____

课程名称 _____数据结构_____ 成绩 _____

一、选择题（每题 2 分，共 40 分）请将答案写在下列表格中

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

1. 一个栈的入栈序列为 1,2,3,. ,n，其出栈序列是 p1, p2, p3,..., pn 。若 p2 = 3，则 p3 可能取值的个数是

A. n-3　　B. n-2　　**C. n-1**　　D. 无法确定

2. 若将关键字 1，2，3，4，5，6，7 依次插入到初始为空的平衡二叉树 T 中，则 T 中平衡因子为 0 的非叶子结点的个数是

A. 0　　B. 1　　C. 2　　**D. 3**

3. 在任意一棵非空二叉查找树 T1 中，删除某结点 v 之后形成二叉查找树 T2，再将 v 插入 T2 形成二叉查找树 T3。下列关于 T1 与 T3 的叙述中，正确的是

I. 若 v 是 T1 的叶结点，则 T1 与 T3 不同

II. 若 v 是 T1 的叶结点，则 T1 与 T3 相同

III. 若 v 不是 T1 的叶结点，则 T1 与 T3 不同

IV. 若 v 不是 T1 的叶结点，则 T1 与 T3 相同

A. 仅 I、III　　B. 仅 I、IV　　**C. 仅 II、III**　　D. 仅 II、IV

4. 对给定的关键字序列 110，119，007，911，114，120，122 进行基数排序，则第 2 趟分配收集后得到的关键字序列是

A. 007，110，119，114，911，120，122　　B. 007，110，119，114，911，122，120

C. 007，110，911，114，119，120，122　　D. 110，120，911，122，114，007，119

我承诺，我将严格遵守考试纪律。

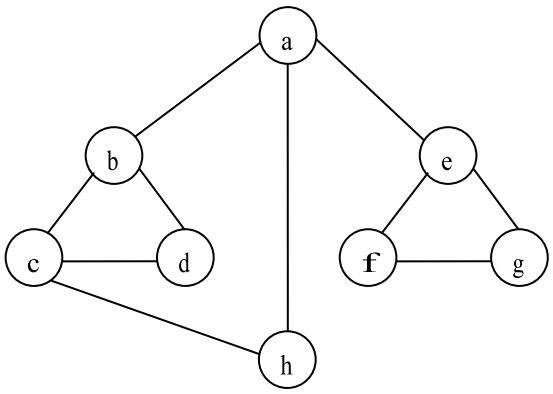
承诺人：_____

题号	一	二	三	四	五	六	七
得分							
批阅人(流水阅卷教师签名处)							

5. 若对如下无向图进行遍历，则下列选项中，不是广度优先遍历序列的是

A. h，c，a，b，d，e，g，f　　B. e，a，f，g，b，h，c，d

C. d，b，c，a，h，e，f，g　　**D. a，b，c，d，h，e，f，g**



6. 在一株高度为 2 的 5 阶 B 树中，根结点所含关键字的个数最少是

A.5　　B. 7　　C. 8　　**D. 1**

7. 已知表头元素为 c 的单链表在内存中的存储状态如下表所示。

地址	元素	链接地址
1000H	a	1010H
1004H	b	100CH
1008H	c	1000H
100CH	d	NULL
1010H	e	1004H
1014H		

现将 f 存放于 1014H 处并插入到单链表中，若 f 在逻辑上位于 a 和 e 之间，则 a，e，f 的“链接地址”依次是

A．1010H，1014H，1004H　　B．1010H，1004H，1014H

C．1014H，1010H，1004H　　**D．1014H，1004H，1010H**

8. 已知一个带有表头结点的双向循环链表 L，结点结构为

prev	data	next

其中，prev 和 next 分别是指向其直接前驱和直接后继结点的指针。现要删除指针 p 所指的结点，正确的语句序列是

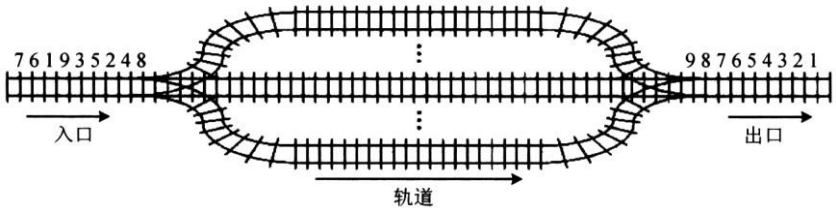
A. p->next->prev=p->prev; p->prev->next=p->prev; delete p;

B. p->next->prev=p->next; p->prev-> next=p->next; delete p;

C. p->next->prev=p->next; p->prev->next=p->prev; delete p;

D. p-> next-> prev=p->prev; p->prev->next=p->next; delete p;

9. 设有如下图所示的火车车轨，入口到出口之间有 n 条轨道，列车的行进方向均为从左至右，列车可驶入任意一条轨道。现有编号为 1～9 的 9 列列车，驶入的次序依次是 8，4，2，5，3，9，1，6，7。若期望驶出的次序依次为 1～9，则 n 至少是



A．2　B．3　**C．4**　D．5

N 个队列， 只能从小到大。 如 9/8 ， 5/4， 3/2，7/6/1

10. 若森林 F 有 15 条边、25 个结点，则 F 包含树的个数是

A．8　B．9　**C．10**　D．11

一个树 24 条边，仅根无入分支。少 1 条边，2 棵树，少 2 条边，3 棵树，现在少 9 条边，故 10 棵树。

11. 若将 n 个顶点 e 条弧的有向图采用邻接表存储，则拓扑排序算法的时间复杂度是

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

12. 在有 n(n>1000)个元素的升序数组 A 中查找关键字 x。查找算法的伪代码如下所示。

```
k=0;
while(k<n 且 A[k]<x)k=k+3;
if(k<n 且 A[k]==x)查找成功;
else if(k-1<n 且 A[k-1]==x)查找成功;
    else if(k-2<n 且 A[k-2]==x)查找成功;
        else 查找失败;
```

本算法与折半查找算法相比，有可能具有更少比较次数的情形是

A．当 x 不在数组中　**B．当 x 接近数组开头处**

C．当 x 接近数组结尾处　D．当 x 位于数组中间位置

While 次数 n/3，后面 2 次比较。共 n/3+2，和 $\log_2 n$ 比较。显然 n/3+2,> $\log_2 n$,所以 x 在开头处 n/3 好

13. B+树不同于 B 树的特点之一是

A．**能支持顺序查找**　　B．结点中含有关键字
C．根结点至少有两个分支　　D．不适合外部文件建立索引用

14. 对 10 TB 的数据文件进行排序，应使用的方法是

A．希尔排序　B．堆排序　C．快速排序　**D．归并排序**

15. 已知程序如下：

```
int S(int n)
{    return(n<=0)?0：s(n-1)+n；    }
void main(  )
{    cout<<S(1)；    }
```

程序运行时使用栈来保存调用过程的信息，自栈底到栈顶保存的信息依次对应的是

A．**main( )→S(1)→S(0)**　　B．S(0)→S(1)→main()
C．main( )→S(0)→S(1)　　D．S(1)→S(0)→main( )

16. 先序序列为 a，b，c 的不同二叉树的个数是

A．3　　**B．5**　　C．4　　D．6

罗列法则，对称的计算一次即可。

17. 下列选项给出的是从根分别到达两个叶结点路径上的权值序列，能属于同一棵哈夫曼树的是

A．24，10，5 和 24，10，7　　B．24，10，5 和 24，12，7
C．24，10，10 和 24，14，11　　**D．24，10，5 和 24，14，6**

18. 现有一棵无重复关键字的平衡二叉树(AVL 树)，对其进行中序遍历可得到一个升序序列。下列关于该平衡二叉树的叙述中，正确的是

A．根结点的度一定为 2　　B．树中最小元素一定是叶结点
C．最后插入的元素一定是叶结点　　**D．树中最大元素一定无右子树**

19. 下列选项中，不能构成折半查找中关键字比较序列的是

A．500，200，450，180　　B．500，450，200，180
C．180，500，200，450　　D．180，200，500，450

20. 下列排序算法中，元素的移动次数与关键字的初始排列次序无关的是

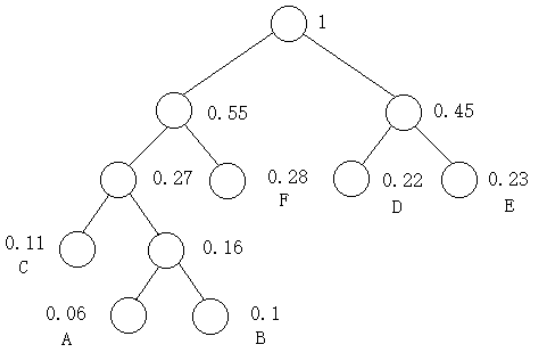
A．直接插入排序　B．冒泡排序　**C．基数排序**　D．快速排序

参考答案：
CDCCD　ADDCC　BBADA　BDDAC

二、简答题（20 分）

（1）已知字符 A、B、C、D、E、F 的相对使用频率为 0.06、0.1、0.11、0.22、0.23、0.28，试根据哈夫曼算法，构造一棵以上述字符为叶结点的二叉树，并给出上述各字符的哈夫曼编码。

参考答案：



A 0010　　B 0011　　C 000　　D 10　　E 11　　F01

（2） 设闭散列表的地址空间为[0,5]，依次插入元素 12，36，41，52，40，散列函数为 H(x)=x mod 6，采用线性探测再散列方式解决冲突，请给出插入后的散列表及查找 40 需要的比较次数。

参考答案：

散列表：

12	36	40		52	41
0	1	2	3	4	5

查找 40 经历了 5 次比较

（3） 一项工程有 A,B,C,D 和 E 五项子工程，各子工程间的施工约束条件为：A>B, A>D, B>E, C>A, C>B, E>D，其中，“A>B”的含义是 A 子工程完成后 B 才可以开始施工，依此类推。请给出该工程可能的施工顺序，要求写明求解思路和过程。

参考答案：

C、A、B、E、D

看哪个结点入度为 0，输出，删除其射出的边，再看哪个结点入度为 0，输出，反复如此，直到输出所有顶点。

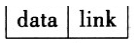
（4） 假设外存的页块大小为 4000 字节，指向外存的地址指针需要 5 字节。现有一个由 $20*10^6$ 条记录构成的文件，每条记录为 200 字节，其中包括关键字 5 字节。请问如果采用 B 树索引文件存储，B 树的阶数应该是多少，要求写明计算过程。

参考答案：

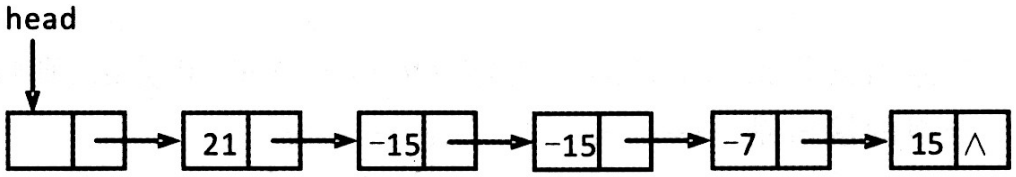
B 树中索引结点和数据结点大小最好是一个页块。

假设 B 树阶数为 M，则每个结点最多有 M 个孩子。当有 M 个孩子时，结点中有 M-1 个关键字、M-1 个记录地址、M 个页块地址。故有 5(M-1)+ 5(M-1)+5M<=4000， M =267。

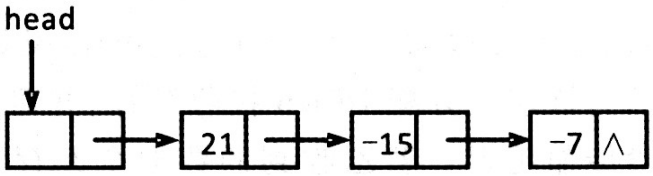
三、程序题（40 分）

1）用单链表保存 m 个整数，结点的结构为： 

，且|data|≤n(n 为正整数，且 m>n)。现要求设计一个时间复杂度尽可能高效的算法，对于链表中 data 的绝对值相等的结点，仅保留第一次的结点而删除其余绝对值相等的结点。例如，若给定的单链表 head 如下：



则删除结点后的 head 为：



要求：

(1)给出算法的基本设计思想。 （5 分）

(2)根据设计思想，采用 C++语言描述算法，关键之处给出注释。 （8 分）

(3)说明你所设计算法的时间复杂度和空间复杂度。 （2 分）

答案

(1)算法的基本设计思想

算法的核心思想是用空间换时间。使用辅助数组记录链表中已出现的数值，从而只需对链表进行一趟扫描。因为|data|≤n，故辅助数组 exist 的大小为 n+1，各元素的初值均为 0。依次扫描链表中的各结点，同时检查 exist[|data|] 的值，如果为 0，则保留该结点，并令 exist[|data|]=1；否则，将该结点从链表中删除。

(2)算法实现

```
void func(Node* head，int n)
{
```

```
        Node *p, *q;
        int *exist; //保存绝对值存在的状态，1 存在，0 不存在

        exist = new int[n+1];
        for (int i=0; i<n+1; i++) exist[i]=0;

        p = head;
        while (p->next)
        {
          if (exist[abs(p->next->data)])
          {  q = p->next;
             p->next = q->next;
             delete q;
          }
          else
          {  exist[abs(p->next->data)]=1;
             p = p->next;
          }
      }
      delete []exist;
}
```

(3)算法的时间复杂度为 O(m)，空间复杂度为 O(n)

2）一个用二叉链表存储的二叉树的根结点地址为 root，结点结构类型为 node{Type data, node*left, node*right}，请设计并实现二叉树类 binaryTree 的一个成员函数。要求该成员函数利用递归方式实现二叉树中所有结点的左右子树交换功能。（10 分）

```
template <class type>
void binaryTree<type>::changLR(node *t)
{   node *temp;
    if (!t) return;
    temp = t->left;   t->left = t->right; t->right = temp;
```

```
    changeLR(t->left);
    changeLR(t->right)
}
```

3） 一个有向图用邻接矩阵存储，其存储结构如下：

```
template <class verType, class edgeType>
class Graph
{
    private:
        int verts, edges;      //图的实际顶点数和实际边数
        int maxVertex;          //图顶点的最大可能数量
        verType *verList;      // 保存顶点数据的一维数组
        edgeType **edgeMatrix; //保存邻接矩阵内容的二维数组
        edgeType noEdge;          //无边的标志，一般图为 0， 网为无穷大 MAXNUM
        bool directed;            //有向图为 1，无向图为 0
    public:
        //初始化图结构 g，direct 为是否有向图标志，e 为无边数据
        Graph(bool direct, edgeType e);
        ~Graph();
        ------------------------------
        bool circleJudge()const;   //图中存在回路返回 true,否则返回 false。
}
```

要求实现函数 circleJudge()const，以判断该有向图是否存在回路并分析算法的时间复杂度。（15 分）

参考代码：

```
template <class verType, class edgeType>
bool Graph< verType, edgeType>:: circleJudge()const
{    //利用拓扑排序算法，检查拓扑排序中的顶点个数
     int *inDegree, count;
     int row, col;
     seqStack<int> s;

     inDegree = new int[verts]; //入度数组
```

```
    //计算入度，入度为零的入栈
    for (col=0; col<verts; col++)
    {  inDegree[col]=0;
       for (row=0; row<verts; row++)
            if (edgeMatrix[row][col]!=noEdge) inDegree[col]++;
       if (inDegree[col]==0) s.push(col);
    }

   //求拓扑序列
   while (!s.isEmpty())
   {     row = s.pop(); count++;
         for ( col = 0; col<verts; col++)
         {       if (edgeMatrix[row][col]!=noEdge) inDegree[col]--;
                 if (inDegree[col]==0) s.push(col);
         }
   }

   //根据序列中的顶点个数判断回路
   if (count==verts) return false;
   return tree;
}
```

求入度时间复杂度 $O(n^2)$，求拓扑序列 $O(n^2)$，总时间复杂度 $O(n^2)$。