

上海交通大学试卷(____卷)

(____至____学年第____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 1 分, 共 15 分, 请将答案填在以下表格里)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- 以下哪个操作在单链表上实现比顺序表实现效率高?
A. 在最后一个元素的后面插入一个新元素 B. 顺序输出前 k 个元素
C. 删除所有值为 x 的元素 D. 交换表中两个元素
- 利用栈求下列后缀表达式的值时, 若设立的操作数栈只有 2 个存储单元, 则不会发生上溢的是?
A. $a b c d - * -$ B. $a b - c * d -$ C. $a b c * - d -$ D. $a b - c d - *$
- 一个容量为 25 的循环队列中, 若头指针 $front=18$, 尾指针 $rear=9$, 则该循环队列中共有多少元素?
A. 16 B. 17 C. 9 D. 10
- 记高度为 h 的二叉树(只含有一个结点的二叉树高度定义为 1)中可能包含的最小结点数为 m , 最大结点数为 M , 则 (m, M) 是?
A. $(\log h, 2^h)$ B. $(\log h, 2^{h-1})$ C. $(h, 2^h)$ D. $(h, 2^{h-1})$
- 对任意一棵由 n 个结点构成的树, 这 n 个结点的度数之和是多少?
A. $n-1$ B. $2n-1$ C. n D. 不确定
- 分块查找方法要求?
A. 块内元素有序, 块间元素无序, 不同块的元素数量需要相等
B. 块内元素可以无序, 块间元素有序, 不同块的元素数量不需要相等
C. 块内元素可以无序, 块间元素有序, 不同块的元素数量需要相等
D. 块内元素有序, 块间元素有序, 不同块的元素数量需要相等
- 一个长度为 n 的序列, 可以构建多个二叉查找树, 其中查找的时间复杂度最坏为?
A. $O(n)$ B. $O(\log n)$ C. $O(\sqrt{n})$ D. $O(n^2)$
- 以下关于哈希表说法错误的是?
A. 采用链地址法解决冲突时, 查找一个元素的时间可能不同
B. 用再散列法解决冲突, 就可以保证不会再发生冲突
C. 采用链地址法解决冲突时, 不会引起再次聚集
D. 采用再散列法解决冲突不需要额外的存储空间

我承诺，我将严格遵守考试纪律。

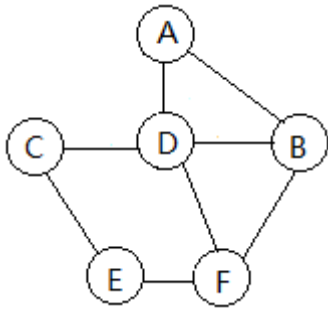
承诺人：_____

题号	一	二	三	四
得分				
批阅人(流水阅卷教师签名处)				

9. 下列排序算法中，在每一趟都能选出一个元素放到其最终位置上，并且其时间性能受数据初始特性影响的是？
A.直接插入排序 B.快速排序 C.直接选择排序 D.堆排序
10. 对一组数据（84，47，25，15，21）排序，数据的排列次序在排序过程中变化如下：
（1）84 47 25 15 21 （2）15 47 25 84 21 （3）15 21 25 84 47 （4）15 21 25 84 47 （5）15 21 25 47 84
则采用的排序方法是？
A. 直接选择排序 B. 冒泡排序 C.快速排序 D.直接插入排序
11. 在一棵5阶B树中有53个关键字，这棵树的高度最高是？（设定B树树高不包括叶结点，只有根结点的树高为1）
A.2 B.3 C.4 D.5
12. 在图的所有基本操作实现算法中，关于时间复杂度，以下哪些说法是正确的？
A. 只和顶点个数有关 B. 和顶点个数、边的条数都无关
C. 只和边的条数有关 D. 可能和顶点个数和边的条数都有关
13. 在外部排序和查找技术中，以下哪个说法是不正确的？
A. B树只含有部分数据的关键字
B. 多阶段归并是为了最大限度地提高归并的路数
C. 利用B+树可以便捷地按关键字递增顺序遍历所有数据
D. 主要的时间消耗是内外存间的数据交换
14. 利用小顶堆实现优先级队列，可使其进队、出队操作的时间复杂度达到：
A. $O(1)$ 和 $O(n)$ B. $O(n)$ 和 $O(\log_2 n)$ C. $O(\log_2 n)$ 和 $O(\log_2 n)$ D. $O(\log_2 n)$ 和 $O(n)$
15. 一棵非空二叉树（n个结点）的先序遍历序列和后序遍历序列正好相反，这棵二叉树的高度必为：
A. n B. $\log_2 n$ C. n+1 D. 1

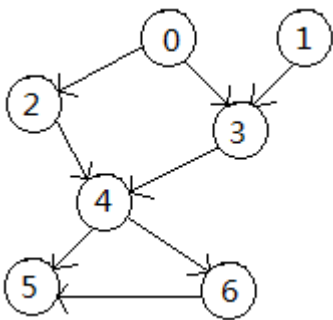
二、 简答题（每题 5 分，共 40 分）

1. 一个无向图如下所示：



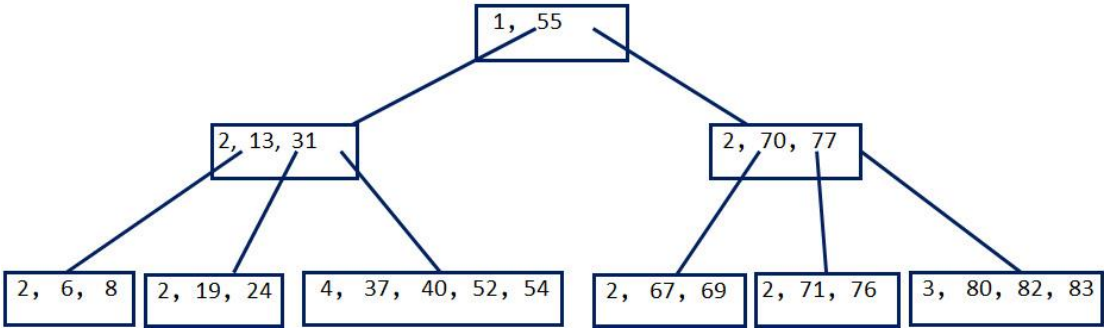
- (1) 该图是否存在欧拉回路？
- (2) 如果有，请画出一个欧拉回路；如果没有，请说明理由。

2. 给出下列有向图所有可能的拓扑序列：



3. 设待排文件 FI={1, 4, 10, 11, 8, 0, 20, 12, 9, 5, 14, 7, 16, 3, 13, 19, 2, 6, 18, 15}, 内存工作区容量为 w=5, 按置换选择法生成的初始归并段为:

4. 如图所示的一棵 5 阶 B 树 (F 叶节点省略), 插入结点 39 后是:

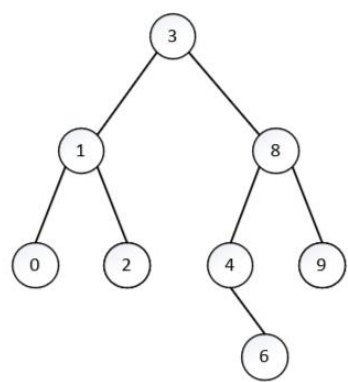


5. 已知待排序序列为{50, 86, 72, 41, 45, 93, 57, 46}, 请写出按下列排序方法进行升序排序时的第一趟排序结果:

- ① 希尔排序 (步长为 4);
- ② 冒泡排序;
- ③ 直接选择排序;
- ④ 二路归并排序;

6. 给出的数据初值为 40, 20, 24, 15, 30, 25, 66, 35, 45, 14, 41, 请给出利用 buildHeap 构造出的最小化堆。

7. 画出在以下 AVL 树中插入 5 后的结果，以及在此结果中再插入 7 后的结果。



8. 已知一个森林的孩子兄弟链表示所对应的二叉树是一棵完全二叉树，该完全二叉树的顺序存储如下图所示：

	A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10	11

请画出这个森林。

三、 程序填空（每空 2 分，共 20 分）

1. //输出二叉树中值为 x 的结点的所有祖先结点的值（假定二叉树中各个结点的值都不相同，且 x 也可能不存在）。

```
template<class elemType>
void binaryTree<elemType>::findAncestors(class elemType x) const
{
    linkStack<Node*> sn; linkStack<int> sf;
    Node *p; int flag;
    bool exist;

    if (!root) { cout<<"x does not exist"<<endl; return; }

    _____
    sn.push(root); sf.push(0);
    while (!sn.isEmpty())
    {   flag = sf.pop();
        p = _____

        if (flag==0) {   sf.puch(1);
                        if (p->left) {sn.push(p->left); sf.push(0); }
                        continue;
                    }
        if (flag ==1) {   sf.puch(2);
                        if (p->right) {sn.push(p->right); sf.push(0); }
                        continue;
                    }
        if (flag ==2) {   sn.pop();
                        if ( p->data ==x)
                        {   exist = true;
                            _____
                        }
                    }
    }

    if (!exist) { cout<<"x does not exist!"; return;}
    if ( _____ ) { cout<<"x does not have any ancestors"; return; }

    cout<<"All of the ancestors of x are:  "
    while (!sn.isEmpty())
    { p = sn.pop();
      cout<< p->data<<"  ";
    }
}
```

2. 以下为在二叉查找树中删除结点的算法实现，请在空白处补充语句。

```
template <class KEY, class OTHER>
```

```
void BinarySearchTree<KEY, OTHER>::remove( const KEY & x, BinaryNode * & t )
```

```
{    if( t == NULL ) _____

    if ( x < t->data.key )    remove( x, t->left );

    else if( t->data.key < x )    remove( x, t->right );

    else    // x = t->data.key

        if ( _____ ) {    //有两个孩子

            BinaryNode *tmp = t->right;

            while ( _____ ) tmp = tmp->left;

            t->data = tmp->data;

            remove( t->data.key, t->right );

        }

    else

        {    //被删结点是叶结点或只有一个孩子

            BinaryNode *oldNode = t;

            t = _____

            delete oldNode;

        }

}
```

3. 以下程序段为优先级队列的入队操作

```
template <class Type>
```

```
void priorityQueue<Type>::enqueue( const Type & x )
```

```
{    if( currentSize == maxSize - 1 )    doubleSpace();

    int hole = ++currentSize;

    for( ; hole > 1 && x < array[ hole / 2 ]; _____ )

        array[ hole ] = array[ hole / 2 ];

    _____

}
```


四、编程题（共 25 分）

1. 一个有向无环图（不带权值）的邻接表类定义如下：

```
template<class typeOfVer, class typeOfEdge>
class adjListGraph
{ private:
    struct edgeNode
    {   int end; edgeNode *next; }
    struct verNode
    { typeOfVer data; edgeNode *head; }

    Int edges, vers; //图中边的条数和顶点的个数
    verNode *verList;
public:
    -----
    int find(typeOfVer x); //返回顶点 x 的下标
    void findAllVertex(typeOfVer source) const;
}
```

请编码实现 findAllVertex(TypeOfVer source)，该函数输出从顶点 source 出发，存在一条路径长度为 M（注：M<e）的所有顶点的值。（8 分）

2.设计一个算法判断一棵二叉树是否为完全二叉树。(8分)

3.有一个国家的外交信件使用如下方法写成密码（密码由 26 个小写字母、空格、标点符号组成）：首先颠倒所有的非元音字母连成的字段序列（包括空格和标点符号，其中元音字母为 a,e,i,o,u），然后再把全部信件倒着写。请你编写一个解密程序翻译信件。

例如：

原文：

data structures

先颠倒所有的非元音字段（包括空格和标点符号）：

datarts utcures

再全部倒着写（最终密码）：

seructu stratad

（9 分）