

上海交通大学试卷(奇卷)

(2021 至 2022 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 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

- 1、 对长度为 3 的顺序表进行顺序查找(从前往后), 查找第一个元素的概率是 $1/4$, 查找第二个元素的概率是 $1/8$, 查找第 3 个元素的概率是 $1/8$, 查找不成功的概率是 $1/2$, 则平均查找长度是:

A. $19/8$ B. $7/8$ C. $7/16$ D. $19/16$

$$\frac{1}{4} + \frac{2}{8} + \frac{3}{8} + \frac{1}{2} = \frac{19}{8}$$

- 2、 对包含 n 个元素的散列表进行查找, 平均时间复杂度:

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

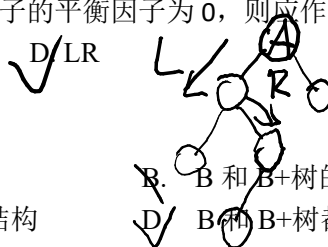
$$1 \times \frac{1}{4} + 2 \times \frac{1}{8} + 3 \times (\frac{1}{8} * \frac{1}{2}) = \frac{19}{8}$$

- 3、 设一组初始记录关键字序列为(57, 345, 53, 67, 49, 162, 247), 则用基数排序需要进行几趟的分配和回收才能使得初始关键字序列变成有序序列?

A. 5 B. 4 C. 3 D. 2

- 4、 在 AVL 树中插入一个结点后造成了不平衡, 设自下层结点往上层结点看, 第一个不平衡结点为 A, 并已知 A 的左孩子的平衡因子为 -1, 右孩子的平衡因子为 0, 则应作哪种类型的调整以使其平衡?

A. RL B. RR C. LL D. LR



- 5、 下列关于 B 和 B+树, 描述不正确的是:

A. B 和 B+树都是平衡的多叉树

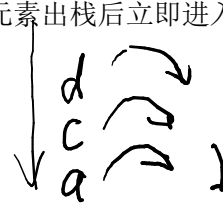
B. B 树和 B+树都可以用于文件索引结构

C. B 和 B+树的数据记录都存储在叶结点上

D. B 和 B+树都能有效地支持随机检索

- 6、 设栈 S 和队列 Q 的初始状态均为空, 元素 a,b,c,d 依次进入栈 S。若每个元素出栈后立即进入队列 Q, 且 4 个元素出队的顺序是 b,d,c,a, 则栈 S 的容量至少是_____。

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



- 7、 设 n 是描述问题规模的非负整数, 下面程序片段的时间复杂度是_____。

```
x=2;
while (x<n)
    x=2*x;
```

$$\lfloor \log_2 n \rfloor$$

A. $O(\log n)$ B. $O(n)$ C. $O(n \log n)$ D. $O(n^2)$

我承诺，我将严格遵守考试纪律。

承诺人：_____

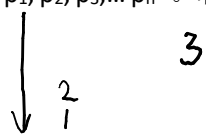
题号	一	二	三	四
得分				
批阅人(流水阅卷教师签名处)				

8、已知两个长度分别为 m 和 n 的升序链表，若将它们合并为一个长度为 $m+n$ 的升序链表，则时间复杂度是_____。

- A. $O(n)$ B. $O(m*n)$ C. $O(m-n)$ ☒ D. $O(m+n)$

9、一个栈的入栈序列为 $1, 2, 3, \dots, n$ ，其出栈序列是 $p_1, p_2, p_3, \dots, p_n$ 。若 $p_1=3$ ，则 $p_2=1$ 且 $p_3=2$ 情况是否可能出现？_____。

- A. 可能 ☒ B. 不可能 C. 无法确定



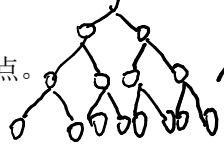
10、循环队列放在一维数组 $A[0..M-1]$ 中， $end1$ 指向队头元素， $end2$ 指向队尾元素的后一个位置。队列中最多能容纳 $M-1$ 个元素，初始时队列为空。下列判断队空和队满的条件中，正确的是_____。

- ☒ A. 队空: $end1 == end2$; 队满: $end1 == (end2+1) \bmod M$
 B. 队空: $end1 == end2$; 队满: $end2 == (end1+1) \bmod (M-1)$
 C. 队空: $end2 == (end1+1) \bmod M$; 队满: $end1 == (end2+1) \bmod M$
 D. 队空: $end1 == (end2+1) \bmod M$; 队满: $end2 == (end1+1) \bmod (M-1)$



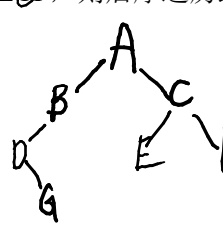
11. 一棵有 5 个叶结点的完全二叉树，最多有多少个结点。

- A. 12 B. 11 C. 9 ☒ D. 10



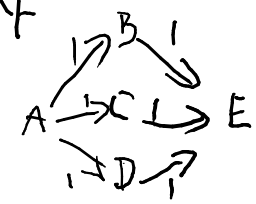
12. 一棵二叉树的前序遍历结果是 $ABDGCE$ ，中序遍历结果是 $DGBAEC$ ，则后序遍历结果是什么？

- A. BDEFGCA ☒ B. GDBEFC C. DBGCEAF D. AGBDEFC



13. 下面关于 AOE 的叙述中，哪个说法是正确的。

- ☒ A. AOE 网是无向图。
☒ B. AOE 网中有且只有一条关键路径，即关键路径具有唯一性。
☒ C. 如果用 AOE 网描述项目工期的话，关键路径上的活动决定了项目的整个工期。
☒ D. AOE 网的关键路径是从起点到终点的加权最短路径。

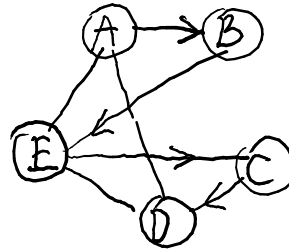


14. 下面关于无向图的叙述中，哪个是不正确的。

- ☒ A. 一个有 n 个顶点和 n 条边的无向图一定是有环的。
☒ B. 用邻接矩阵存储图时所需存储空间大小与图的结点数和边数都有关。
☒ C. 所有顶点的度之和为偶数。
☒ D. 用邻接表法存储图所用的空间大小与图的顶点数和边数有关。

15. 无向图 G 的邻接矩阵及顶点存储顺序表如下，其中，顶点在邻接矩阵中的下标顺序与其在顺序表中的存储顺序相对应。对该图从 A 点开始，按照深度优先遍历算法进行遍历，得到的顶点序列正确的是哪个。

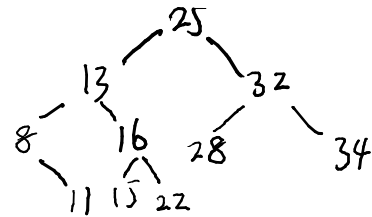
	A	B	C	D	E
A	0	1	0	1	1
B	1	0	0	0	1
C	0	0	0	1	1
D	1	0	1	0	1
E	1	1	1	1	0



0	1	2	3	4
A	B	C	D	E

ABECD

- A. ABCDE B. ACBDE ☒ C. ABECD D. ABDEC



二、简答题（每题 5 分，共 40 分）

1. 从空树开始，依次插入元素 25, 16, 32, 13, 28, 8, 11, 22, 34 和 15 后构成的 AVL 树是怎样的？
2. 设待排文件 $FI=\{1, 27, 7, 3, 75, 44, 12, 15, 8, 10\}$ ，内存工作区容量为 $w=3$ ，按置换选择法生成的所有初始归并段是什么？
 $1, 3, 7, 27, 44, 75, 8, 10, 12, 15$
 $3, 7, 27, 44, 75, 12, 8, 12, 15$
3. 序列 $\{17, 15, 26, 37, 45, 66, 4, 14\}$ ，利用快速排序算法从小到大进行排序，以第一个关键字为基准，得到的第一趟划分结果序列是怎样的？
 $14, 15, 4, 17, 45, 66, 37, 26$
 $\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

4. 请说明线性表、栈、队列这三种数据结构的共同点与区别。

同：①线性 ②存储（顺序/链式） ③操作
 异：①线性（FILO, FIFO） ②操作 ③复杂度

5. 用单链表保存 m 个整数，结点的结构为：

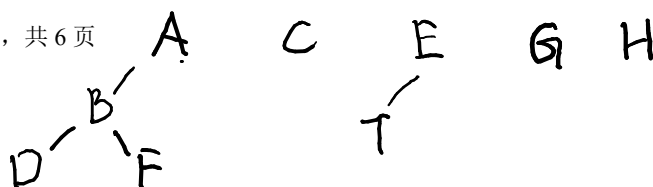
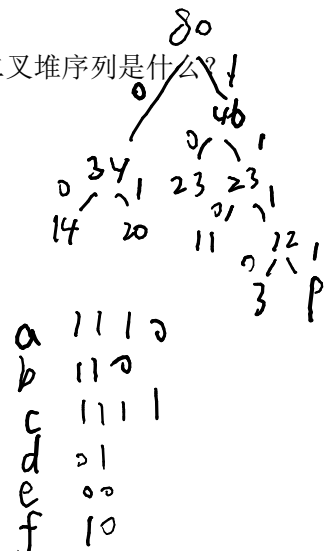
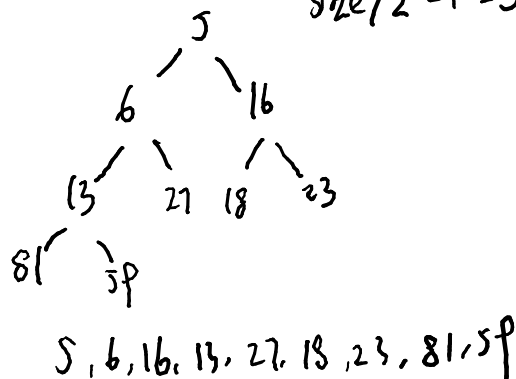
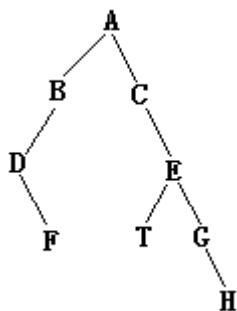
data	next
------	------

且 $|data| \leq n$ (n 为正整数)。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，仅保留第一次出现的结点而删除其余绝对值相等的结点。给出算法的基本设计思想并说明你所设计算法的时间复杂度和空间复杂度。

大小为 n 的链表，遍历 $list$ ，查找 $|x|$ ，若不存在，插入 $|x|$ ，并保存 x ，否则无操作
 6. 已知字符 a, b, c, d, e, f 的权值分别为 3, 11, 9, 20, 14, 28，请画出为这六个字符编码的哈夫曼树。
 时间 $O(m)$ ，空间 $O(n)$

7. 已知关键字集合 $\{18, 81, 5, 13, 27, 16, 23, 6, 59\}$ ，则初始化的最小化二叉堆序列是什么？

8. 若用二叉树存储森林，请画出以下二叉树对应的森林。



三、 程序填空（每空 2 分，共 20 分）

1. 以下是二叉查找树插入的非递归实现，设结点类有 3 个数据成员：data、left 和 right，并有构造函数 Node(T x, Node *l=NULL, Node *r=NULL)。空格处应当填写的正确选项是（ ）。

```
template <class T>
```

```
void binarySearchTree<T>::insert(T x)
```

```
{
```

```
    if ( root == NULL ) {
```

```
        root = new Node(x);
```

```
        return;
```

```
    }
```

```
    Node *t = root;
```

```
    while (true) {
```

```
        if (x == t->data)
```

```
            return;
```

```
        if ((x < t->data && t->left == NULL) || (x > t->data && t->right == NULL))
```

```
            break;
```

```
        else if (x > t->data)
```

```
            t = t->right;
```

```
        else
```

```
            t = t->left;
```

```
    }
```

```
    if ( t->left == NULL )
```

```
        t->left = new Node(x);
```

```
    else
```

```
        t->right = new Node(x);
```

```
}
```

2. 已知如下代码的功能是检测字符串 str 中的圆括号是否匹配，则空白处的代码分别是？

```
bool CheckMatch(const char* str)
```

```
{ char ch;
```

```
  seqStack<char> stk;
```

```
  while (ch = *str++)
```

```
  { switch (ch)
```

```
    {
```

```
      case '(':
```

```
        stk.push(ch)
```

```
        break;
```

```
      case ')':
```

```
        if ( stk.empty() ) return false;
```

```
        stk.pop();
```

```
        break;
```

```
    }
```

```
  }
```

```
  return true;
```

```
  return stk.empty();
```



```
}
```

3. 已知如下代码的功能是在带头结点单链表中查找第一个值在 $[\min, \max]$ 间的元素位置，则空白处的代码分别是？假设首结点中元素满足条件则返回 0。

```
template<class elemType>
```

```
int sLinkList<elemType>::findInDomain(const elemType &min, const elemType &max)
```

```
{ int i = 0;
  node *p = head -> next
  while(p)
  {
    if( min <= p->data && p->data <= max )
      return i;
    i++;
    p = p->next;
  }
  return -1;
}
```

head 为头结点

4. 用邻接表存储图的顶点和边的结点类定义如下：

```
struct edgeNode { //存储边的结点类
    int end;      //终点
    edgeNode *next;
    edgeNode(int e, edgeNode *n = NULL) { end = e; next = n;}
};
```

```
struct verNode { //存储顶点的结点类
    TypeOfVer ver; //顶点值
    edgeNode *head; //对应的单链表的头指针
    verNode( edgeNode *h = NULL) { head = h;}
};
```

请补充代码，完成图的拓扑排序。

```
template <class TypeOfVer, class TypeOfEdge>
```

```
void adjListGraph<TypeOfVer, TypeOfEdge>::topSort( ) const
```

```
{ linkQueue<int> q;
  edgeNode *p;
  int current, *inDegree = new int[Vers];
  for (int i = 0; i < Vers; ++i) inDegree[i] = 0;
```

```
  for ( i = 0; i < Vers; ++i)
    for (p = verList[i].head; p != NULL; p = p->next)
      ++inDegree[p->end];
```

```
  for (i = 0; i < Vers; ++i) if (inDegree[i] == 0) q.push(i);
```

```
  cout << "拓扑排序为: " << endl;
```

```

while( !q.isEmpty() ){
    current = q.dequeue( );
    cout << verList[current].ver << '\t';
    for (p = verList[current].head; p != NULL; p = p->next)
        if( inDegree[p->end] ) q.enqueue( p->end );
    }
    cout << endl;
}

```

delete[] ?

四、 编程题（共 25 分）

1. 编写函数 Node* find(int value, Node*t), 实现在二叉查找树上进行查找的递归算法。
(10 分)

```

struct Node {
    int data;
    Node* left;
    Node* right;
};

```

2. 二叉树的结点类说明如下:

```

struct Node {    //二叉树结点
    Node  *left , *right ;
    T data;
    Node() : left(NULL), right(NULL) { }
    ~Node() {}
};

```

编写函数 int binaryTree<T>::height (binaryTree<T>::Node *t) const 求给定二叉树的高度，只包含根结点的二叉树高度定义为 1。(10 分)

3. 已知一个带有表头结点的单链表，结点结构为： struct Node{ int data, Node *next}
假设该链表只给出了头指针 head。在不改变链表的前提下，请设计一个尽可能高效的算法，查找链表中倒数第 k 个位置上的结点(k 为正整数)。
若查找成功，函数输出该结点的 data 域的值，并返回 1；否则，返回 0。(5 分)

程序题

第一题

```
Node *find(int value, Node *t)
{
    if (t->data == value)
        return t;
    if (value < t->data)
        return find(value, t->left);
    else
        return find(value, t->right);
}

find(x, root);
```

第二题

```
int max(int x, int y)
{
    return x < y ? y : x;
}

template <typename T>
int binaryTree<T>::height(binaryTree<T>::Node *t) const
{
    if (t == nullptr)
        return 0;
    return 1 + max(height(t->left), height(t->right));
}
```

C++

第三题

```
bool find(Node *head, int k)
{
    Node *p1 = head, *p2 = head;
    for (int i = 1; i <= k; i++)
    {
        p2 = p2->next;
        if (i < k && p2 == nullptr)
            return 0;
    }
    while (p2 != nullptr)
    {
        p1 = p1->next;
        p2 = p2->next;
    }
    std::cout << p1->data;
    return 1;
}
```