

@考试必备

备考专用资料

科学规划内容—系统复习
备考题库训练—题海战术
多重模拟测试—强化记忆
高频考点汇编—精准高效
历年真题演练—考前冲刺

注：下载资料前请认真核对、仔细预览，确认无误后再点击下载。

祝您逢考必过，成功上岸，一战成名

河北建筑工程学院

二〇一九年硕士研究生入学考试自命题试卷 A

考试科目代码 808 考试科目名称 计算机专业基础综合

(注意：所有答案必须写在答题纸上，做在试卷或草稿纸上无效)

重要提示：专业考试为 1+1 (3 选 1) 模式，共 150 分，其中数据结构(试卷一)90 分为必做题。计算机组成原理(试卷二)60 分、计算机网络(试卷三)60 分、计算机操作系统(试卷四)60 分，考生任选其中一个科目作答。在以下选做的科目的方框内打勾。并在答题纸上注明所选做的试卷。

计算机组成原理 计算机网络 计算机操作系统

试卷一：数据结构（必做）

一、单选题(在每小题的四个备选答案中选出一个正确答案，每小题 2 分，共 20 分)

1. 数据结构是一门研究非数值计算的程序设计问题中计算机的数据元素以及它们之间的()和运算等的学科。

A. 结构 B. 关系 C. 运算 D. 算法

2. 若一个线性表中最常用的操作是取第 i 个元素和找第 i 个元素的前趋元素，则采用()存储方式最节省时间。

A. 顺序表 B. 单链表 C. 双链表 D. 单循环链表

3. 在一个链队列中，front 和 rear 分别为头指针和尾指针，则插入一个结点 s 的操作为()。

A. $front=front->next;$ B. $s->next=rear;rear=s;$
C. $rear->next=s;rear=s;$ D. $s->next=front;front=s;$

4. 设 a, b 为一棵二叉树上的两个结点，在中序遍历时， a 在 b 前面的条件是()。

A. a 在 b 的右方 B. a 在 b 的左方 C. a 是 b 的祖先 D. a 是 b 的子孙

5. 假定在一棵二叉树中，度为 2 的结点数为 15，度为 1 的结点数为 30，则叶子结点数为() 个。

A. 15 B. 16 C. 17 D. 47

6. 在一个包含 n 个顶点的有向图中，如果所有顶点的出度之和为 s ，则所有顶点的入度之和为()。

A. n B. s C. $2n$ D. $s-1$

7. 下列关于图遍历的说法不正确的是()。

A. 连通图的深度优先搜索是一个递归过程
B. 图的广度优先搜索中邻接点的寻找具有“先进先出”的特征

- C. 非连通图不能用深度优先搜索法
D. 图的遍历要求每一顶点仅被访问一次
8. 对一棵二叉排序树按()遍历, 可得到结点值从小到大的排列序列。
A. 先序 B. 中序 C. 后序 D. 层次
9. 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()。
A. 顺序查找 B. 折半查找 C. 哈希查找 D. 分块查找
10. 若需要在 $O(n\log_2 n)$ 的时间内完成对数组的排序, 且要求排序是稳定的, 则可选的排序方法是()。
A. 快速排序 B. 堆排序 C. 归并排序 D. 直接插入排序

二、填空题(每个空 2 分, 共 20 分)

11. 数据结构的逻辑结构可分为_____①_____两大类。
12. 设单链表的结点结构为(data, next)。已知指针 p 指向单链表中的结点, q 指向新结点, 欲将 q 插入到 p 结点之后, 则需要执行的语句: _____②_____。
13. 设循环队列的容量为 70, 现经过一系列的入队和出队操作后, front 为 20, rear 为 11, 则队列中元素的个数为: _____③_____。
14. 采用二叉链表作为二叉树的存储结构, 则具有 n 个结点的二叉树有_____④_____个空链域。
15. 在赫夫曼树中, 结点的度数可能取值是: _____⑤_____。
16. 若以邻接矩阵表示有向图, 则邻接矩阵上第 i 行中非零元素的个数即为顶点 v_i 的_____⑥_____。
17. 判定一个有向图是否存在回路, 可以利用_____⑦_____。
18. 有一个有序表为 {1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100}, 当折半查找值为 82 的结点时, _____⑧_____次比较后查找成功。
19. 具有相同函数值的关键字对哈希函数来说称为_____⑨_____。
20. 直接插入排序在关键字正序有序的情况下, 其排序时间复杂度为_____⑩_____。

三、解答题(共 30 分)

21. 已知栈的基本操作函数:

```
int InitStack(SqStack *S); //构造空栈
int StackEmpty(SqStack *S); //判断栈空
int Push(SqStack *S, ElemType e); //入栈
int Pop(SqStack *S, ElemType *e); //出栈
```

函数 conversion 实现十进制数转换为八进制数, 请将函数补充完整。

```
void conversion() {
    InitStack(S);
    scanf("%d", &N);
    while(N) {
        _____(1)_____ ;
        N=N/8;
    }
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/718013054131006077>