

# 南京审计大学

## 2021 年硕士研究生招生考试初试（笔试）试题（ A 卷 ）

科目代码: 814

满分: 150 分

科目名称: 数据结构

注意: ①认真阅读答题纸上的注意事项; ②所有答案必须写在答题纸上, 写在本试题纸或草稿纸上均无效; ③本试题纸须随答题纸一起装入试题袋中交回!

### 一. 名词解释 (共 6 小题, 每题 5 分, 共 30 分)

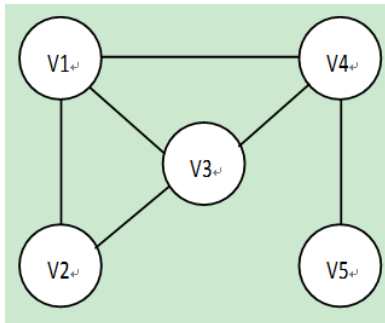
1. 物理结构
2. 数据类型
3. 算法
4. 二叉平衡树
5. 十字链表
6. 关节点

### 二. 简答题 (共 4 小题, 每题 10 分, 共 40 分)

1. 结构是指数据元素之间的相互关系。请简要描述常用基本结构。
2. 请简述希尔排序基本原理。
3. 请简述哈希表的基本工作原理。
4. 一棵二叉树, 叶子结点数为  $n_0$ , 度为 2 的结点数为  $n_2$ 。证明等式成立:  $n_0 = n_2 + 1$

### 三. 综合应用题 (共 8 小题, 每题 10 分, 共 80 分)

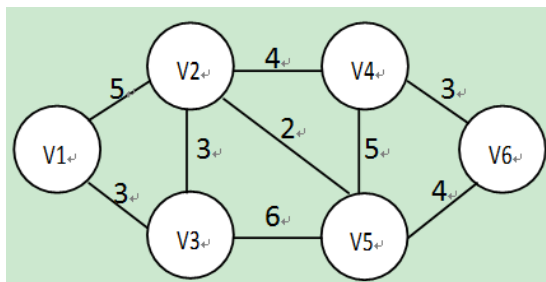
1. 已知二叉树中序遍历序列为 ABCDEFGH; 后序遍历序列为 BADCGHFE. 请写出前序遍历序列。
2. 已知无向图 G, 如图所示。请画出其邻接矩阵, 并写出从 V1 开始深度优先遍历的结点序列。



3. 假设南京审计大学 2021 级学生共 7 个专业, 每个专业的招生人数是: A 300、B 240、C 200、D 180、E 160、F 120、G 100。请为每个专业赋予唯一 2 进制编码, 使学生专业编码的总存储空间占用最少。

4. 已知初始关键字序列 54、94、59、80、53、78、28、19、64。请以 54 作为枢轴关键字, 演示第一趟快速排序的交换过程。

5. 已知带权无向连通图, 如图所示。用图表演示普里姆算法从 V2 起始构造最小生成树过程。



6. 已知有向无环图 AOE 网 G，边代表活动，采用邻接表存储，其拓扑排序后顶点存储于 T 栈，数组 ve[i] 记录每个顶点事件 i 的最早开始时间。请把如下计算关键路径的算法补充完整。

Status CriticalPath (AOEGraph G)

```
{
    vl[0..G.vexnum-1]=ve[0..G.vexnum-1]; //初始化顶点最迟开始时间
    while(!StackEmpty(T))
        for(Pop(T, j)), p=G.vertices[j].firstarc; p; p=p->nextarc)
            {
                k=p->adjvex;
                dut=*(p->info); //耗时权重
                if(vl[k]-dut<vl[j]) _____(1)_____ ;
            }

    for(j=0; j<G.vexnum; ++j)
        {
            for(p=G.vertices[j].firstarc; p; p=p->nextarc)
                {
                    k=p->adjvex;
                    dut=*(p->info);
                    _____(2)_____
                    _____(3)_____
                    tag=(ee==el)?'*': '.';
                    printf( j, k, dut, ee, el, tag); //输出关键活动
                }
        }
}
```

7. 已知 La 和 Lb 是按元素值递增排列的单链线性表。写算法把 La 和 Lb 合并为 Lc，保持数据递增排列规律，并分析其时间复杂度。

void Merge\_List( LinkList &La, LinkList &Lb, LinkList &Lc)

```
{
    }
```

8. 请把计算二叉树深度函数 TreeDepth() 补充完整。

typedef struct BinaryTree

```
{
    int data;
    struct BinaryTree *Left;
    struct BinaryTree *Right;
    struct BinaryTree *Parent;
}Node;
```

//递归求二叉树深度

int TreeDepth(BinaryTree\* root)

```
{
}
```