

武汉科技学院

二 00 四年招收硕士学位研究生试卷

试卷代号 412 试卷名称 管理信息系统

考试时间 2004 年 1 月 11 日 下午 报考专业 管理科学与工程

1、试题内容不得超过画线范围，试题必须打印，图表清晰，标注准确。

2、试题之间不留空格，版面不够时，请接背面，不另加纸。

| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 十一 | 得分 |
|----|---|---|---|---|---|---|---|---|---|---|----|----|
| 得分 | | | | | | | | | | | | |

一、名词解释（3 分*5=15 分）

- 1、数据处理 2、计算机网络 3、管理信息系统 4、信息技术 5、区间码

二、填空（3 分*8=24 分）

- 常用的数据模型有：_____。
- 模块分解的原则是：_____。
- 系统划分的常用方法有：_____。
- CMM 就是软件的 _____ 模型，可分为五个级别_____。
- DSS 就是一个交互的人机系统，它利用 _____、_____ 和方法库，以及很好的人机会话部件和图形部件，帮助决策者进行 _____ 或 _____ 决策的系统。
- 诺兰（Nolan）关于信息系统发展所经历的六个阶段是：_____。
- 模块之间联结方式有：_____。
- MIS 开发的四种方式有：_____。

三、选择题（单选或多选，2×8=16）

- 电子数据处理系统（EDPS）可分为的阶段是：

A、单项数据处理阶段和综合数据处理阶段 B、单项数据处理和 MIS

C、综合数据处理、MIS 和 DSS D、MIS
- 模块设计及评价原则是

A、块间联系最小 B、块间联系最大

C、块内联系最小 D、块内联系最大
- 软件维护的类型有：

A、正确性维护 B、适应性维护

C、完善性维护 D、预防性维护
- 结构化系统分析的主要工具有：

} ListNode;

(1) 已知指针 p 所指结点不是尾结点，若在 *p 之后插入结点 *s，则应执行下列哪一个操作？

- A. s->link = p; p->link = s; B. s->link = p->link; p->link = s;
 C. s->link = p->link; p = s; D. p->link = s; s->link = p;

(2) 非空的循环单链表 first 的尾结点（由 p 所指向）满足：

- A. p->link == NULL; B. p == NULL;
 C. p->link == first; D. p == first;

八、（8分）设有一个顺序栈 S，元素 s1, s2, s3, s4, s5, s6 依次进栈，如果 6 个元素的出栈顺序为 s2, s3, s4, s6, s5, s1，则顺序栈的容量至少应为多少？

九、（10分）一棵具有 n 个结点的理想平衡二叉树（即除离根最远的最底层外其他各层都是满的，最底层有若干结点）有多少层？若设根结点在第 0 层，则树的高度 h 如何用 n 来表示（注意 n 可能为 0）？

十、（10分）从供选择的答案中选择与下面有关图的叙述中各括号相匹配的词语，将其编号填入相应的括号内。

(1) 对于一个具有 n 个结点和 e 条边的无向图，若采用邻接表表示，则顶点表的大小为（ A ），所有边链表中边结点的总数为（ B ）。

(2) 采用邻接表存储的图的深度优先遍历算法类似于树的（ C ）。

(3) 采用邻接表存储的图的广度优先遍历算法类似于树的（ D ）。

(4) 判断有向图是否存在回路，除了可以利用拓扑排序方法外，还可以利用（ E ）。

供选择的答案

A: ① n ② n+1 ③ n-1 ④ n+e B: ① e/2 ② e ③ 2e ④ n+e

C~D: ① 中根遍历 ② 先根遍历 ③ 后根遍历 ④ 按层次遍历

E: ① 求关键路径的方法 ② 求最短路径的 Dijkstra 方法

③ 深度优先遍历算法 ④ 广度优先遍历算法

十一、（10分）设带表头结点的双向链表的定义为

```
typedef int ElemType;
typedef struct dnode { //双向链表结点定义
    ElemType data; //数据
    struct dnode * lLink, * rLink; //结点前驱与后继指针
} DblNode;

typedef DblNode * DblList; //双向链表
```

试设计一个算法，改造一个带表头结点的双向链表，所有结点的原有次序保持在各个结点的右链域 rLink 中，并利用左链域 lLink 把所有结点按照其值从小到大的顺序连接起来。

十二、（14分）下面是求连通网络的最小生成树的 Prim 算法的实现，中间有 5 个地方缺失，请

阅读程序后将它们补上。

```
const int MaxInt = INT_MAX; //INT_MAX 的值在中
const int n = 6; //图的顶点数, 应由用户定义
typedef int AdjMatrix[n][n]; //用二维数组作为邻接矩阵表示
typedef struct { //生成树的边结点
    int fromVex, toVex; //边的起点与终点
    int weight; //边上的权值
} TreeEdgeNode;
typedef TreeEdgeNode MST[n-1]; //最小生成树定义
void PrimMST ( AdjMatrix G, MST T, int rt ) {
//从顶点 rt 出发构造图 G 的最小生成树 T, rt 成为树的根结点
TreeEdgeNode e; int i, k = 0, min, minpos, v;
for ( i = 0; i < n; i++ ) //初始化最小生成树 T
if ( i != rt ) {
T[k].fromVex = rt;
T[k].toVex = i;
T[k].weight = G[rt][i];
k++;
}
for ( k = 0; k < n-1; k++ ) { //依次求 MST 的候选边
min = MaxInt;
for ( i = k; i < n-1; i++ ) //遍历当前候选边集合
if ( T[i].weight < min ) //选具有最小权值的候选边
{ min = T[i].weight; minpos = i; }
if ( min == MaxInt ) //图不连通, 出错处理
{ cerr << "Graph is disconnected!" << endl; exit(1); }
e = T[minpos]; T[minpos] = T[k]; T[k] = e;
v = T[k].toVex;
for ( i = k+1; i < n-1; i++ ) //修改候选边集合
if ( G[v][i] < T[i].weight ) {
T[i].weight = G[v][i];
T[i].fromVex = v;
}
}
}
}
```