

数据结构

试题 (共 7 页)

考生注意:

本试卷共 11 题, 其中第一、二、三、六、七、八题为参加全国统一考试的考生必答题; 第一、四、五、九、十、十一题为参加“单独考试”的考生必答题。

写算法的语言可以用类 PASCAL 语言, 也可以用某一种程序设计语言, 但不允许采用生僻古怪的表示方法。

一、(本题共 30 分, 每小题 6 分)

简要回答下列问题:

1. 一个完整的算法应该具有哪几个性质(特征)?

正确, 有确定性, 输出, 有限(时间空间), 清晰, 有穷性, 有穷性

2. 有人说一个 $m \times n$ 阶矩阵是一个广义表结构, 你认为如何? 请说明理由。

3. 带权连通图中源点到图中其他各顶点的最短路径构成一棵带权生成树, 该生成树是否是该图的最小生成树? 为什么?

4. 何谓索引文件? 索引文件中的索引表通常包含哪些信息? 索引表有何特点? 索引表中包含一个记录的关键字值与记录在文件中的地址。

5. 用文字简述在 B+ 树上查找一个记录存在与否的过程。

二、(本题共 10 分, 每小题 2 分)

已知 AOE 网的逆邻接表如题二图一所示,

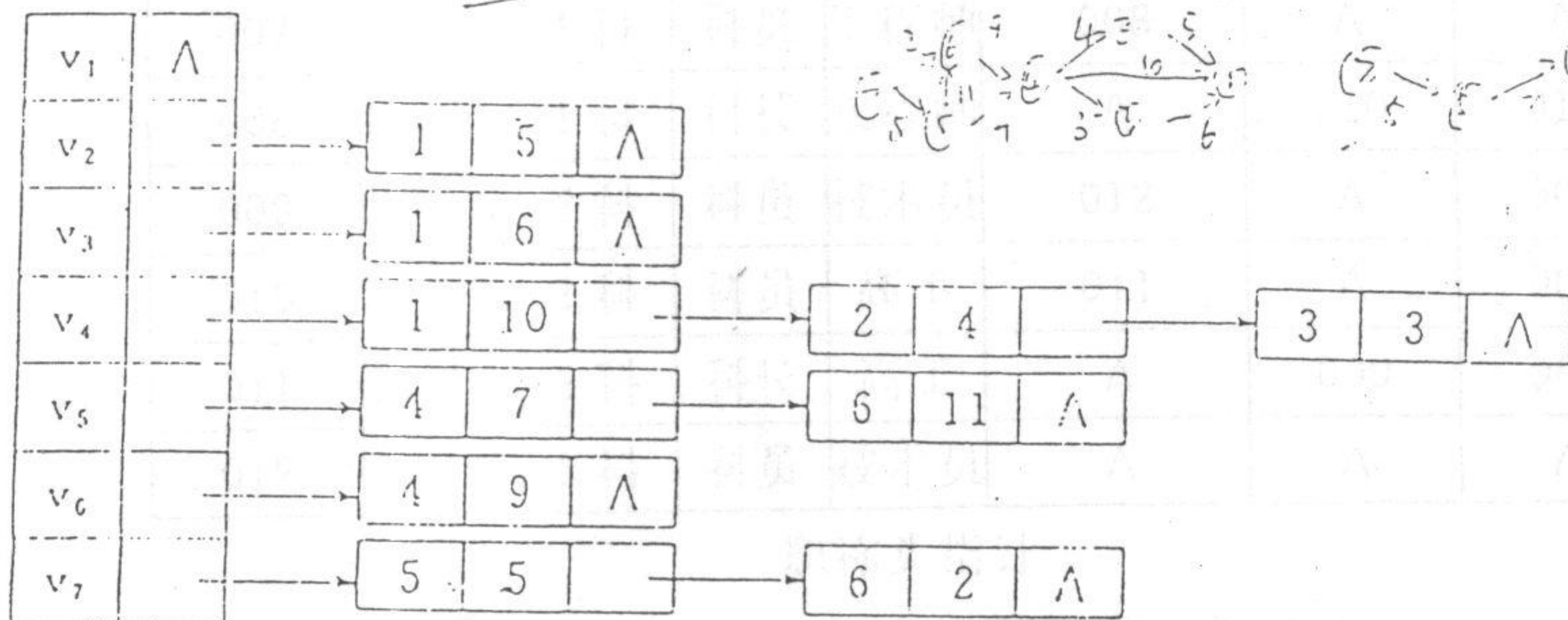
1. 请画出这个 AOE 网;

2. 分别计算出网中每个事件的最早发生时间 $ee[i]$;

3. 分别计算出网中每个事件的最迟发生时间 $le[i]$;

4. 分别计算出每个活动的松弛时间 $diff[i]$;

5. 指出关键路径。



三、(本题 15 分)

图三给出某公司关于员工在计算机中存放的信息, 其中, 除了员工的基本信息外, 还建立了各种指针, 从而表达出员工之间的联系。请问: 题三图一中使用了哪几种基本的存储结构, 每种存储结构对应的逻辑结构, 每种逻辑结构能反映员工之间的什么联系? (指针域指向某职工的职工号, 符号 Λ 表示该指针值为空)

	1	2	3	4	5	6	7	8	9	10
ee	0	5	6	10	30	19	35			
le	0	6	7	10	30	19	35			
$diff$										

职工号	姓 名	单 位	职 务	职 称	职称指针	左指针	右指针
001	刘 成	3 科	科员	工程师	007	Λ	002
002	赵 军	3 科	科员	高工	003	Λ	012
003	李 原	2 科	科员	高工	004	Λ	Λ
004	马 宁	公司办	总经理	高工	006	008	Λ
005	张 英	1 科	科员	技术员	009	Λ	007
006	王 凯	3 科	科长	高工	010	001	Λ
007	高 明	1 科	科员	工程师	008	Λ	Λ
008	田 云	1 科	科长	工程师	Λ	010	011
009	周 华	2 科	科员	技术员	012	Λ	003
010	陈 喜	1 科	科员	高工	011	Λ	005
011	魏 贞	2 科	科长	高工	Λ	009	006
012	奚 宇	3 科	科员	技术员	Λ	Λ	Λ

职称头指针

根指针

004

高工

工程师

技术员

002	001	005
-----	-----	-----

题三图一

四、(本题 10 分)

某航空公司在六个城市设有分公司 v_1, v_2, v_3, v_4, v_5 和 v_6 , 矩阵 A 中元素 $A[i, j]$ 表示 v_i 到 v_j 的直航飞机票价, ($A[i, j] = \infty$ 表示 v_i 到 v_j 的不直接通航)。试为该航空公司制作一张由 v_1 到各分公司去的最便宜的通航线路图。带权连通图的最小生成树

	1	2	3	4	5	6
1	0	50	∞	40	25	10
2	50	0	15	20	∞	25
3	∞	15	0	10	20	∞
4	40	20	10	0	10	25
5	25	∞	20	10	0	55
6	10	25	∞	25	55	0

五、(本题 15 分)

已知动态存储管理的内存释放算法 FREE(AV), 内存可用区的链表如题五图一所示。其中每个链结点的构造如题五图二所示, LLINK 为左指针域, RLINK 为右指针域, TAG 为标志位; 当 TAG=0, 表示本块为自由块, TAG=1, 表示本块为已分配块, SIZE 为块的大小, UPLINK 为向上指针, 它给出本块的首地址。请按题中给出的算法 FREE(AV), 在题五图一给出的初始状态基础上先释放题五图三的块, 再释放题五图四的块, 分别画出两次释放后的可用区的链表。

```
procedure FREE(AV)
```

```
  n ← SIZE(p)    // p 为释放块的首地址 //
```

```
  case
```

```
    : (TAG(p+n)=0 and (TAG(p-1)=1):
```

```
      RLINK(LLINK(p+n)) ← p
```

```
      LLINK(RLINK(p+n)) ← p
```

```
      LLINK(p) ← LLINK(p+n)
```

```
      RLINK(p) ← RLINK(p+n)
```

```
      SIZE(p) ← n + SIZE(p+n)
```

```
      UPLINK(p+SIZE(p)-1) ← p
```

```
      TAG(p) ← 0
```

```
      if AV=p+n then AV ← p
```

```
    : (TAG(p+n)=1) and (TAG(p-1)=0):
```

```
      q ← UPLINK(p-1)
```

```
      SIZE(q) ← SIZE(q) + n
```

```
      UPLINK(p+n-1) ← q
```

```
      TAG(p+n-1) ← 0
```

```
  : else:
```

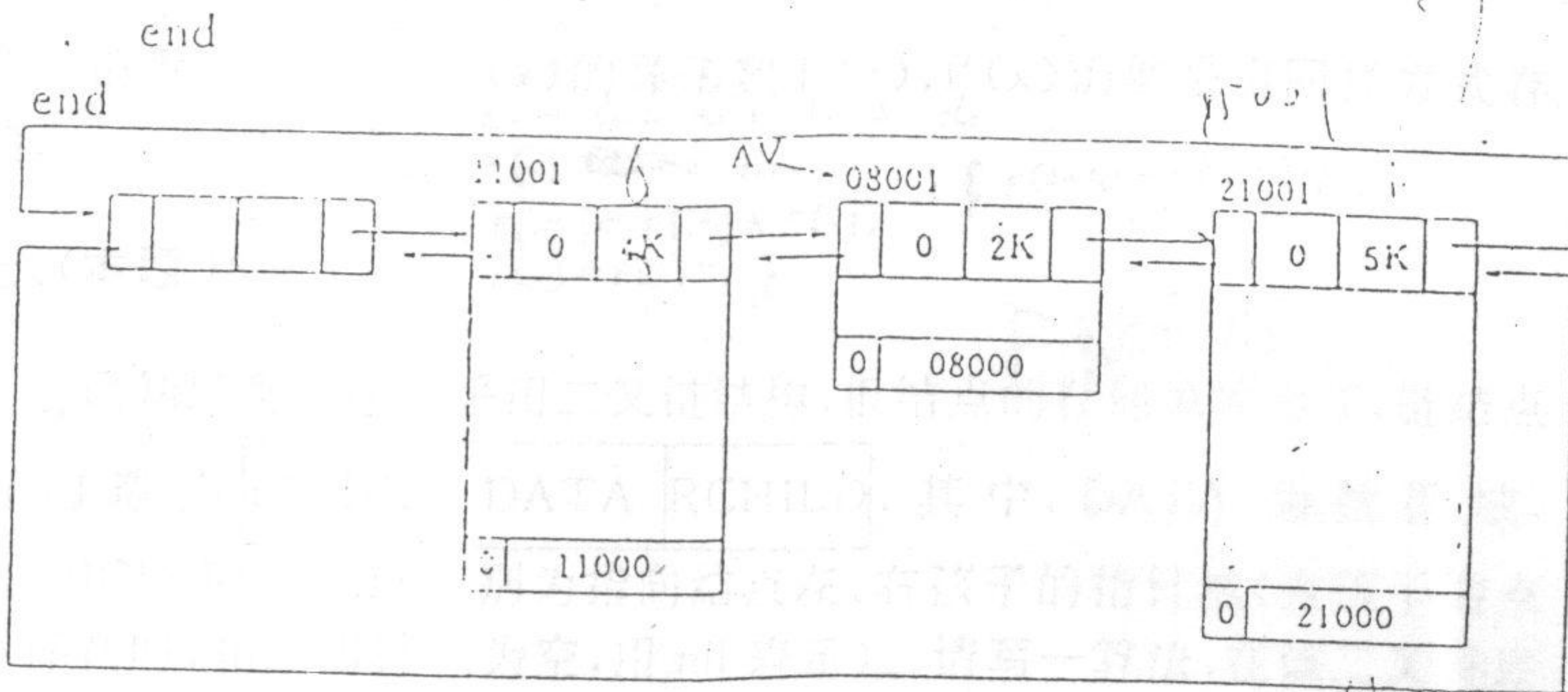
与后邻块合并

与前邻块合并

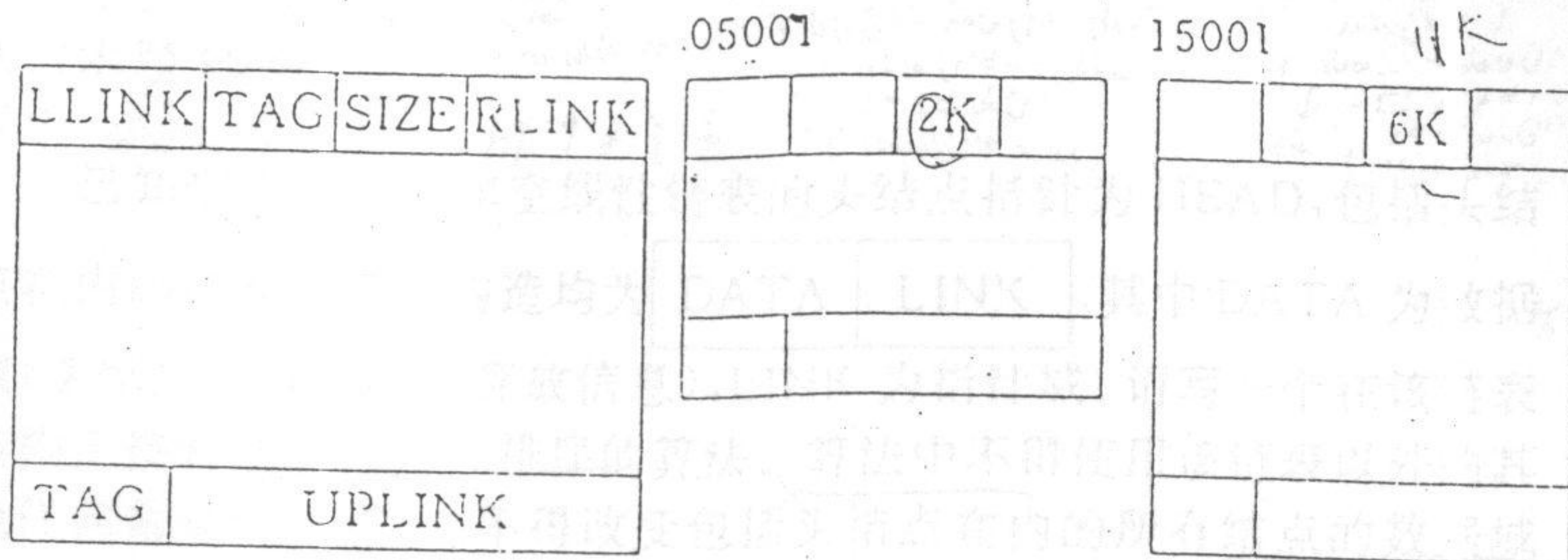

```

TAG(p) ← TAG(p) + 1
UPLINK(p + n - 1) ← p
LLINK(p) ← AV
RLINK(p) ← RLINK(AV)
LLINK(RLINK(p)) ← p
RLINK(AV) ← p
    
```

插入一个结点



题五图一



题五图二

题五图三

题五图四

六、(本题 10 分)

已知一元 n 阶多项式函数 $f(x)$ 由 m 个系数非零项组成, 并采用如

下方法存储: 定义一个一维数组 $F[1: 2m+1]$, 其中 $F[1]$ 存放 m , $F[2]$ 至 $F[2m+1]$ 分别存放多项式函数中各项的“系数, 指数”偶对。例如, $f(x) = 6x^4 - 8x^2 + 5x$, 有

3	6	4	-8	2	5	1
---	---	---	----	---	---	---

请写一算法, 求 $f(x)$ 的导函数 $f'(x)$, $f'(x)$ 仍然采用同样方式存储。

for $i \leftarrow 2m+1$ downto 1
 if $F[i] \neq 0$ then
 { $F[2m+1-i] \leftarrow F[2m+1-i] * i$;
 $F[2m+2-i] \leftarrow F[2m+2-i] - 1$;
 }
 else

七、(本题 15 分)

已知二叉排序树采用二叉链结构, 根结点的存储地址为 T , 链结点的构造为

LCHILD	DATA	RCHILD
--------	------	--------

, 其中: DATA 为数据域, LCHILD、RCHILD 分别为指向结点左、右孩子的指针域(当孩子结点不存在时, 相应指针域为空, 用 nil 表示)。请写一算法, 在该二叉排序树中查找结点数据域值为 key 的结点, 若查找成功, 则给出被查到结点的存储地址, 否则将 key 插入该二叉排序树。

八、(本题 20 分)

已知带头结点的非空线性链表的头结点指针为 HEAD, 包括头结

点在内的所有结点的构造均为

DATA	LINK
------	------

, 其中 DATA 为数据域(头结点的数据域未存放信息), LINK 为指针域。请写一个在该链表结构上按升序进行选择排序的算法。(算法中不得使用该链表以外的其他任何附加链结点, 也不得改变包括头结点在内的所有结点的数据域内容。)

九、(本题 10 分)

设某循环链表长度大于 1, 并且无头结点和头结点指针, 已知 p 为指向该链表中某结点的指针, 链结点的构造为

DATA	LINK
------	------

。请写

算法,删除并替换中缀表达式的直接前驱结点。

十、(本题 15 分)

已知中缀表达式存放于字符数组 $E[1:MAX]$ 中, (设表达式中每个运算对象都用一个字母或数字表示), 并且以 '@' 作为表达式的结束标志。请写一算法, 判断该中缀表达式中左、右括号是否配对, 并给出相应信息。

利用栈

遇到左括号

压入栈

遇到右括号

Top = 0 则结束

十一、(本题 20 分)

已知二叉树采用二叉链存储结构, 根结点的存储地址为 T, 链结点的构造为

LCHILD	DATA	RCHILD
--------	------	--------

, 其中 DATA 为数据域, LCHILD、RCHILD 分别为指向结点左右孩子的指针域 (当孩子结点不存在时, 相应指针域为空, 用 nil 表示), 请写一非递归算法, 判断该二叉树是否是二叉排序树, 并给出相应信息。

利用栈

按层遍历

