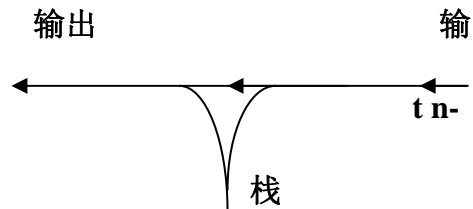


## 数据结构期终考卷（计算机系 99）

姓名\_\_\_\_\_ 班级\_\_\_\_\_ 学号\_\_\_\_\_ 实验班\_\_\_\_\_

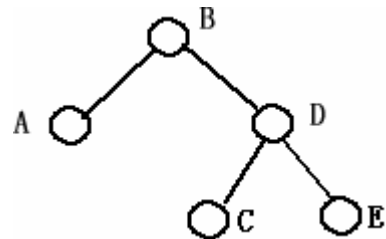
### 1 回答下列问题（15 分）

- (1) 对哈希(HASH)函数  $H(k) = k \text{ MOD } m$ , 一般来说  $m$  应取  
A) 素数 B) 很大的数 C) 偶数 D) 奇数
- (2) 设计一个判断表达式中左右刮号是否配对的算法, 采用\_\_\_\_\_数据结构最佳.  
A) 队列 B) 堆栈 C) 二叉树 D) 链表
- (3) 一个  $n$  个顶点的连通无向图, 其边的个数至少为:  
A)  $n-1$  B)  $n$  C)  $n+1$  D)  $n \log n$
- (4) 当字符序列  $tn-$  作为右图输入  
入 时, 输出长度为 3 的, 且可用作  
C 语言标识符的序列的有.  
A) 4 个 B) 5 个 C) 3 个 D) 6 个
- (5) 在 **deap** 中左子树 (min-heap) 上的节点  $i$ , 与其相对应的右子树 (max-heap) 中的节点  $j$  等于  
A)  $i + 2^{\lfloor \log_2 i \rfloor - 1}$  B)  $i - 2^{\lfloor \log_2 i \rfloor - 1}$  C)  $i + 2^{\lfloor \log_2 i \rfloor + 1}$  D)  $i - 2^{\lfloor \log_2 i \rfloor + 1}$

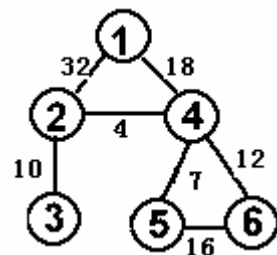


### 2. 画出下列各图（12 分）

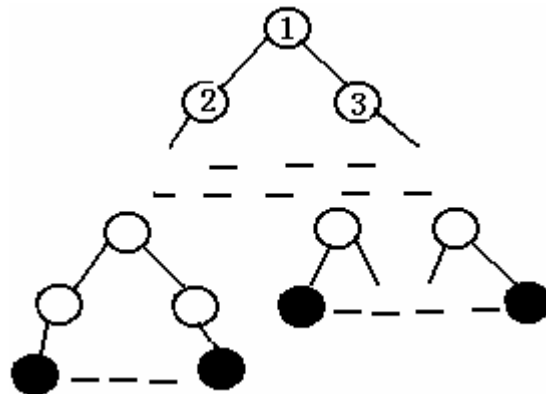
- (1) 画出右图的 AVL 树中插入节点“F”后的 AVL 树（4 分）



- (2) 对右图所示的无向图画出最小成本生成树。（4 分）



(3) 下列完全二叉树共有  $d$  层及  $n$  个节点，试在下图涂黑的节点（叶节点）标上相应的序号。（4 分）



3. 假定标识符 A、B、C、D 在应用中的使用频率分别为 0.4、0.3、0.2、0.1，请试用 **huffman tree** (a binary tree with minimum weighted external path length) 的方法，对这些标识符进行 2 进制编码。（12 分）

1) 在下面画出 **huffman tree** （8 分）

2) 填写下列的标识符 2 进制编码（4 分）

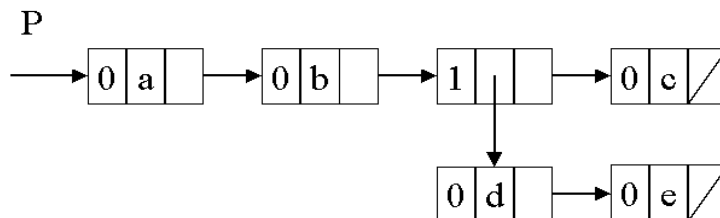
标识符	2 进制编码
A	
B	
C	
D	

4. 广义表的结点结构如右图所示：

其中 LINK 为表中指向下一元素的指针，

TAG	DATA	LINK
-----	------	------

TAG 为标志域，TAG=0 表示该结点为原子，DATA 为数据；TAG=1 表示该结点为一子表，DATA 为指向子表的指针。下图为一广义表的例子。



说明如下算法的功能，当输入为上图时，画出其输出结果。（10 分）

**PROCEDURE** A(p, t) /\* 算法中 p、t、m、n、r、q 均为指针\*/

**BEGIN**

q := NIL;

**WHILE** p <> NIL **DO**

**BEGIN**

**IF** p^.TAG <> 0 **THEN**

**BEGIN**

m := p^.DATA

A(m, n);

p^.DATA := n

**End;**

r := p^.link;

p^.link := q;

q := p;

p := r

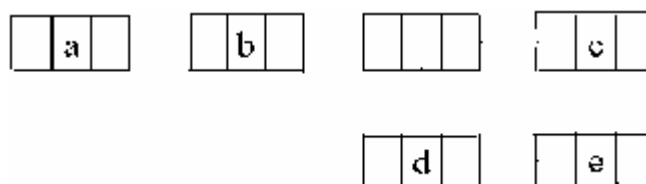
**End;**

t := q

**END**

算法功能\_\_\_\_\_（5 分）

输出结果图（5 分）



5. 下面是将任意序列调整为最大堆(max heap)的算法, 将空白部分填上: (10 分)

将任意序列调整为最大堆通过不断调用 adjust 函数, 即:

```
for (I = n/2; I > 0 ; I--) adjust(list, I, n );
```

其中 list 为待调整序列所在数组 (从下标 1 开始), n 为序列元素个数, adjust 函数为:

```
void adjust(int list[], int root, int n)
```

/\* 将以 root 为下标的对应元素作为待调整堆的根, 待调整放在 list 数组中, 最大元素下标为 n\*/

```
{ int child, rootkey;
  rootkey=list[root];
  child = 2* root;
  while (child<=n) {
    if ((child<n) && (list[child]<list[child+1]))
      _____;
    if (rootkey > list[child])
      break;
    else {
      list [_____] = list[child];
      child *= 2;
    }
  }
  _____=rootkey;
}
```

6. 判断下列序列能否构成最大堆 ( 12, 70, 33, 65, 24, 56, 48, 92, 86, 35); 若不是, 则按上述算法将其调整为堆, 调整后的结果为: (6 分)

(\_\_\_\_\_)

7. 考察一个二处理机的调度问题, 各作业所需的不同处理机的时间由矩阵 J 给出

$$J = \begin{bmatrix} 3 & 6 \\ 4 & 2 \\ 8 & 9 \\ 10 & 15 \end{bmatrix} \quad \begin{matrix} T1 \\ T2 \\ T3 \\ T4 \end{matrix}$$

根据动态规划法进行最优调度, 写出其四个作业的处理次序及完成所有作业的时间 (10 分)

作业的处理次序 \_\_\_\_\_ (4 分)

完成所有作业的时间 \_\_\_\_\_ (6分)

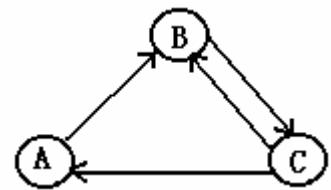
8. 已知下图中, 图节点的构造为

发出的弧表	进入的弧表	图点信息	下一图点
-------	-------	------	------

弧节点构造为

起始节点	终止节点	下一条同起始节点的弧	下一条同终止节点的弧
------	------	------------	------------

这种结构表示图的优点是当删除一个节点时, 可以同时删除其发出的和进入的所有弧。试画出右图在该节点类型下的节点连接图 (10分)



9. 试在 Binary Search Tree 中删去数据域小于  $x$  的节点 ( $x$  为正数, 且  $1 \leq x \leq 100$ ), 二叉树的数据域为  $1 \sim 100$  的正整数 (15分)

1) 编写该算法 (7分)

2) 编写 C 语言程序实现该算法 (8分)