

2012-2013(1)数据结构 A 类笔试答案

一、(每题 1.5 分, 共 30 分)

1. C 2. A 3. C 4. C 5. C
6. D 7. A 8. B 9. D 10. A
11. D 12. C 13. D 14. C 15. D
16. C 17. D 18. A 19. C 20. D

二、(每题 1.5 分, 共 30 分)

1. (3 分)

`n-i; r[j+1]=r[j];`

2. (3 分)

`return t; t=t->rchild;`

3. (10.5 分)

`0; j; i; 0; inDegree[i]==0; edge[vex][i]==1; inDegree[i]==0;`

4. (3 分)

`hashtable[i]=0, hashtable[k]=s`

5. (10.5 分)

`length; flag=0; i+=2; t=a[i+1]; i+=2; t=a[i+1]; flag;`

三、(20 分)

1. (6 分)

1) 从 while (所剩边数 \geq 顶点数) 来看, 循环到边数比顶点数少 1 (即 $n-1$) 停止, 这符合 n 个顶点的连通图的生成树有 $n-1$ 条边的定义; (2 分)

2) 由于边是按权值从大到小排序, 删去的边是权值大的边, 结果的生成树必是最小生成树; (2 分)

3) 算法中“若图不再连通, 则恢复 e_i ”, 含义是必须保留使图连通的边, 这就保证了是生成树, 否则或者是有回路, 或者成了连通分量, 均不再是生成树。(2 分)

2. (6 分)

交换 (3 分) 二叉树的左右子树 (3 分) 的递归算法

3. (8 分)

排序算法的思想是: (1 分)

第一趟比较将最小的元素放在 $r[1]$ 中, (2 分)

最大的元素放在 $r[n]$ 中, (2 分)

第二趟比较将次小的放在 $r[2]$ 中,

将次大的放在 $r[n-1]$ 中,

依次下去, (1 分)

直到待排序列为递增序。 (2 分)

四、(12 分)

1. (4 分)

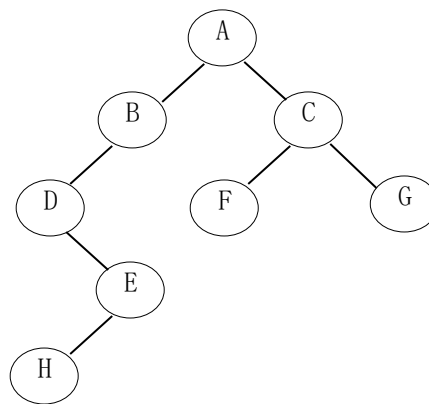
`struct node {` (1 分)

```

    int year, month, day;
};
typedef struct {           (3分)
    int num;               //帐号
    char name[8];          //姓名
    struct node date;      //开户年月日
    int tag;               //储蓄类型, 如: 0- 零存, 1- 一年定期……
    float put;             //存入累加数
    float interest;        //利息
    float total;           //帐面总数
} count;

```

2. (8分)



(3分)

1) 按前序遍历: 根左右, 则第一个元素必为树的根节点, 如上例中的 A 就为根节点

(1分)

2) 按中序遍历: 左根右, 按根节点 A 得知, 左子树包含元素 DHEB, 右子树包含元素 FCG

(1分)

3) 按前序分解为 DHEB 和 FCG (根被消去了)

4) 递归进行左子树求解 (左子树的中序为 DHEB, 前序为 FCG)

(1分)

5) 递归的进行右子树的求解 (右子树的中序为 DHEB, 前序为 FCG)

(1分)

6) 直至递归到没有左右子树为止

(1分)

五、(8分)

```

void quickpass( int r[], int s, int t ) {
    int i = s, j = t, x = r[s];   (1分)
    while ( i < j ) {             (2分)
        while ( i < j && r[j] > x ) j = j-1; if ( i < j ) { r[i] = r[j]; i = i+1; }
        (2分)
        while ( i < j && r[i] < x ) i = i+1; if ( i < j ) { r[j] = r[i]; j = j-1; }
        (2分)
    }
}

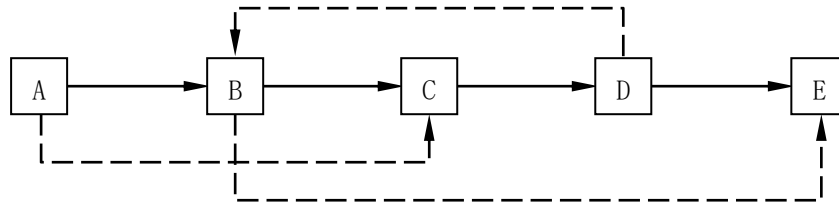
```

```

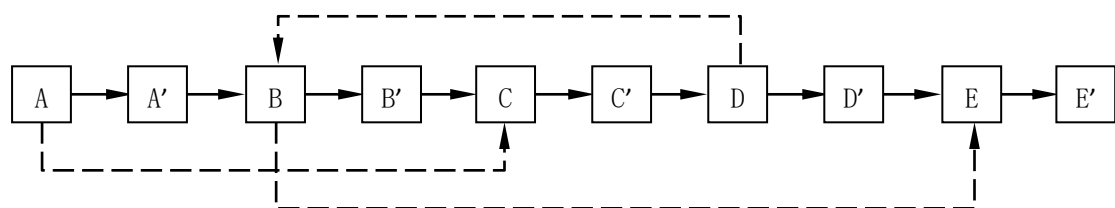
    }
    r[i]=x;    (1分)
}

```

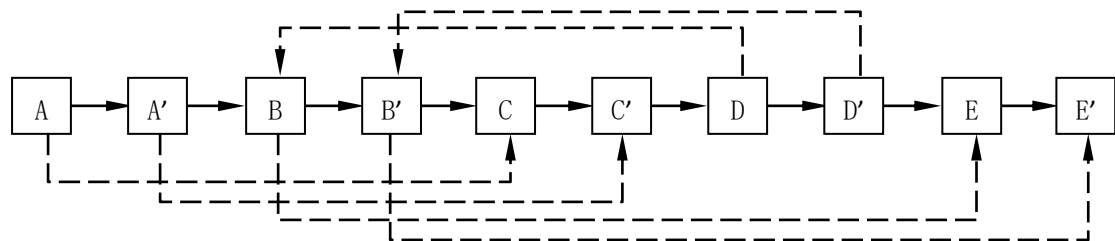
六、(10分)



1. 根据原始链表的每个结点 N 创建对应的 N'，并把 N' 链接在 N 的后面。(4分)



2. 设置复制出来的结点的 m_pSibling。设原始链表上的 N 的 m_pSibling 指向结点 S，那么其对应复制出来的 N' 的 m_pSibling 指向结点 S'。(3分)



3. 把上述长链表拆分成两个链表，奇数位置的结点用 m_pNext 链接起来就是原始链表，偶数位置的结点用 m_pNext 链接起来就是复制出来的链表。(3分)

