



厦门大学《数据结构》期末试题·答案

考试日期：2006.1 (cyj)

信息学院自律督导部



一、简答题（请与出王步骤或埋田）

20 分

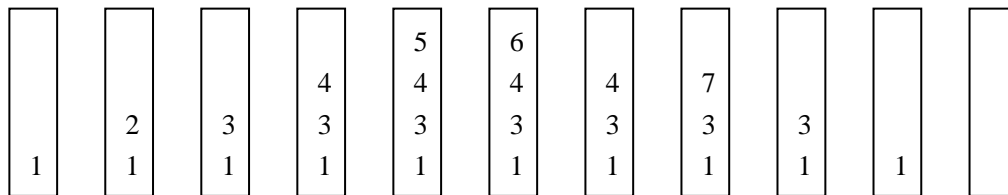
- (1) N 个结点的二叉树，最大、最小深度各为多少？
- (2) n 个结点 ($n > 0$) 的完全二叉树，有多少个叶子结点？
- (3) 数据 1,2,3,4,5,6,7 依次入顺序栈，出栈序列为 2,5,6,4,7,3,1，栈的数组长度至少要多少？
- (4) 设树 T 的度为 4，其中度为 1, 2, 3 和 4 的结点个数分别为 4, 2, 1, 1 则 T 中的叶子数为多少？
- (5) 在双向循环链表中,向 p 所指的结点之后插入指针 f 所指的结点，其操作是 $f \rightarrow \text{prior} = p$; _____; $f \rightarrow \text{next} = p \rightarrow \text{next}$; _____。

解答：

(1) N 个结点的二叉树，最大深度为 n，最小深度为 $\lfloor \log_2 n \rfloor + 1$

(2) 编号为 n 的结点的父结点的编号为 $\lfloor n/2 \rfloor$ ，该结点是最后一个非终端结点，即从编号为 1 到编号 $\lfloor n/2 \rfloor$ 的结点均是非终端结点，所以叶子结点的个数为 $n - \lfloor n/2 \rfloor$ 。

(3) 栈的变化过程如下：



当 5 或 6 入栈时，栈中的元素个数最多，所以栈的数组的长度至少为 4。

(4) 假设树 T 的边数为 B，结点数为 n，则有 $B = 0 \cdot n_0 + 1 \cdot n_1 + 2 \cdot n_2 + 3 \cdot n_3 + 4 \cdot n_4$

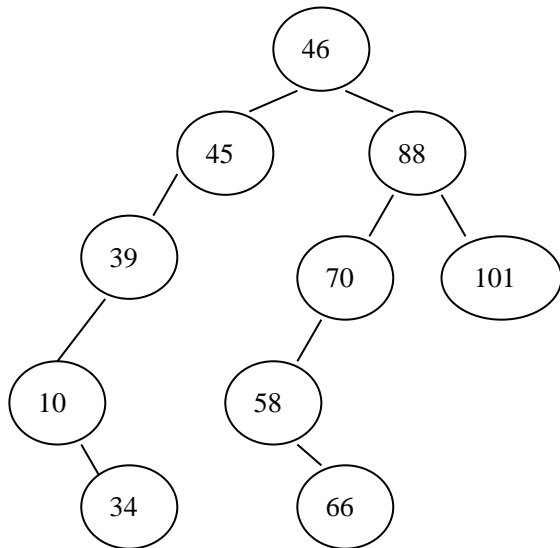
$$B = n - 1, \quad n = n_0 + n_1 + n_2 + n_3 + n_4, \quad n_1 = 4, \quad n_2 = 2, \quad n_3 = 1, \quad n_4 = 1$$

则有 $B = 15, n = 16$, 所以树 T 中的叶子数为 $n_0 = 8$

(5) $p \rightarrow \text{next} \rightarrow \text{prior} = f$; $p \rightarrow \text{next} = f$;

二、用序列(46,88,45,39,70,58,101,10,66,34)建立一个排序二叉树，画出该树，并求在等概率情况下查找成功的平均查找长度。

解答：



等概率下平均查找长度 $ASL = (1*1 + 2*2 + 3*3 + 4*2 + 5*2) / 10 = 3.2$

三、编写递归函数判断二叉树 T 是否是满二叉树，假设以二叉链表存储。

```

typedef struct BiTNode{
    TElemType data;
    struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
  
```

解答：

满二叉树中任一结点为根的子树都是满二叉树。

算法：

- (1) 如果二叉树 T 是空树，则是满二叉树；
- (2) 如果二叉树 T 非空，左右子树都是满二叉树，而且深度一样，则 T 是满二叉树；
- (3) 如果二叉树 T 非空，左子树或右子树不是满二叉树，则不是满二叉树；
- (4) 如果二叉树 T 非空，左右子树都是满二叉树，但深度不一样，则 T 不是满二叉树。

//该函数判断二叉树 T 是否是满二叉树

//如果是满二叉树，返回 TRUE，Depth 返回该树的深度；

//否则返回 FALSE，Depth 无定义；

Boolean Check(BiTree T, int &Depth)

```

{ int ldepth, rdepth;
  if( T==NULL) { Depth=0; return TRUE; }
  if( Check(T->lchild, ldepth)==FALSE ) return FALSE;
  if( Check(T->rchild, rdepth)==FALSE) return FALSE;
  if( ldepth!=rdepth ) return FALSE;
  Depth=ldepth+1; return TRUE;
}
  
```

四、一棵二叉树的先序、中序和后序序列分别如下，其中有一部分未显示出来，试求出空格处的内容，并

画出该二叉树。

先序序列：__B__F__ICEH__G；

中序序列：D__KFIA__EJC__；

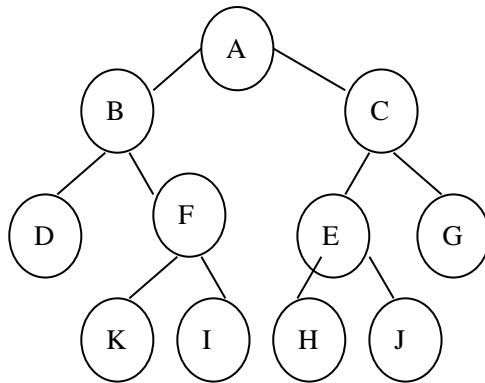
后序序列：__K__FBHJ__G__A。

解答：先序序列为 ABDFKICEHJG；

中序序列为 DBKFIAHEJCG；

后序序列为 DKIFBHJEGCA。

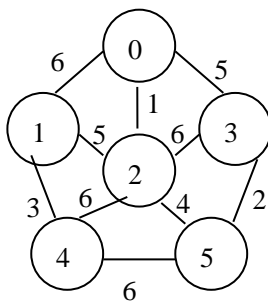
二叉树如下：



五、下面是带权无向图 G。

(1) 从顶点 0 出发，写出深度优先遍历序列和广度优先遍历序列，当有多种选择时，编号小的结点优先。

(2) 分别使用普里姆算法和克鲁斯卡尔算法求下图的最小生成树，仅需画出最小生成树的成长过程。

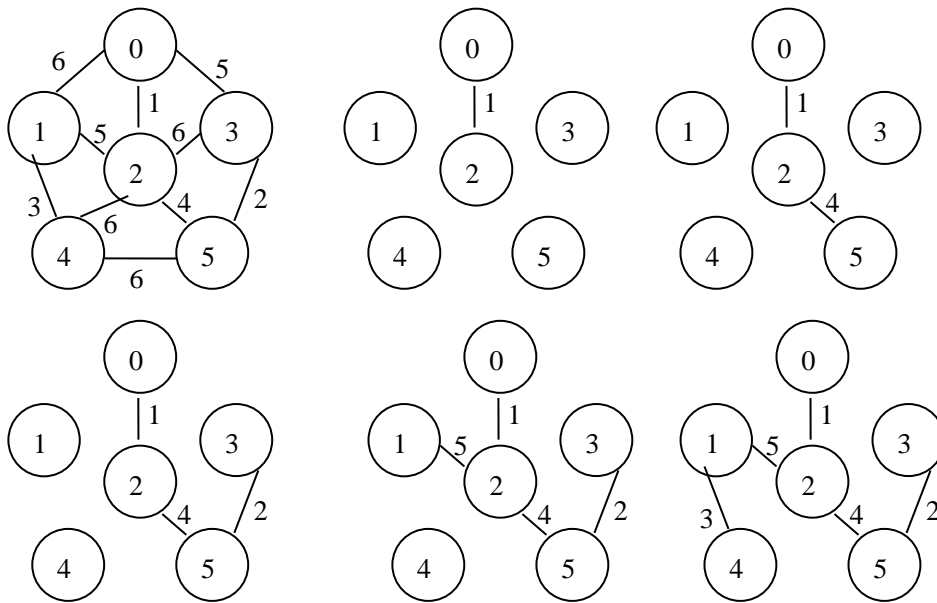


解答：

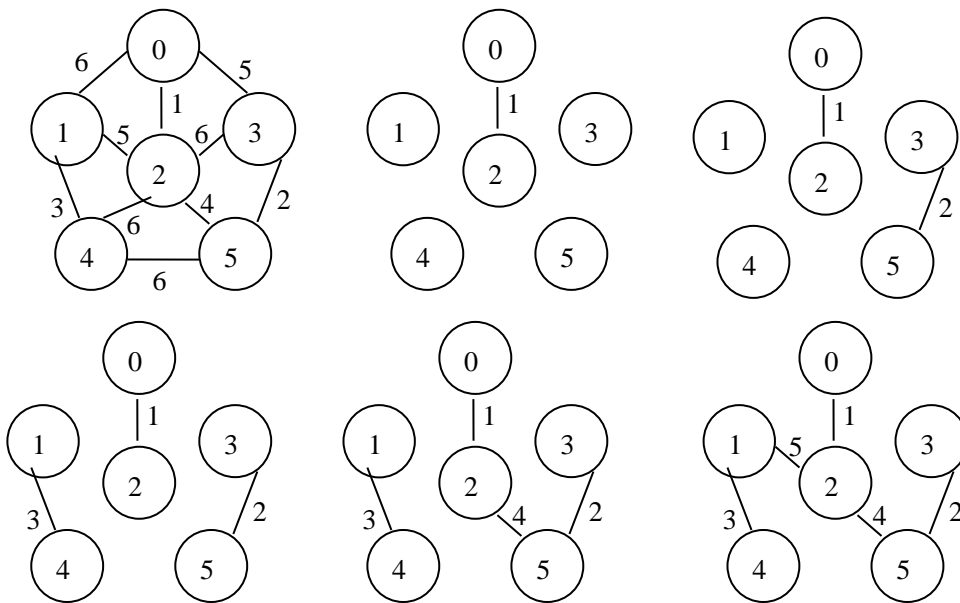
(1) 深度优先遍历序列为：0，1，2，3，5，4

广度优先遍历序列为：0，1，2，3，4，5

(2) 普里姆算法求最小生成树的过程如下：



克鲁斯卡尔算法求最小生成树的过程如下：



六、设待排序的表有 8 个记录，其关键字分别为：18，2，20，34，12，32，6，16。写出用 2--路归并排序的每趟结果。2-路归并排序是否是稳定的？

解答：

	(18)	(2)	(20)	(34)	(12)	(32)	(6)	(16)
一趟归并后	(2 18)	(20 34)	(12 32)	(6 16)				
二趟归并后	(2 18 20 34)	(6 12 16 32)						
三趟归并后	(2 6 12 16 18 20 32 34)							

2—路归并排序是稳定的。