

2005春季学期试题 参考答案:

一、填空:

1. 2, 1.4 (7/5) 2. 38,46,56,79,40,80, 3. $O(\log_2 n)$, $O(n\log_2 n)$ 4. 出度,
入度 5. (0,1),(1,3),(3,2),(1,4)或5,3,6,8 6. 55 7.5

二、选择:

- 1C, 2B, 3B, 4B, 5B, 6C, 7B, 8C, 9A, 10B

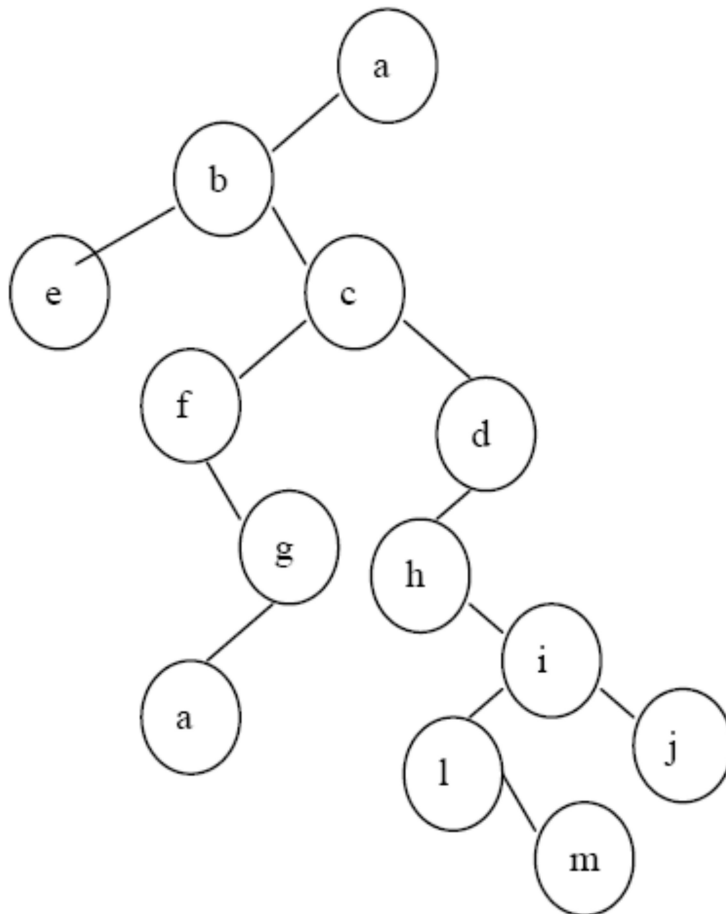
三、判断正误:

- 1√, 2√, 3×, 4√, 5×, 6×, 7×, 8×, 9×, 10×

四、简答题:

1. 一个栈模拟入队操作, 一个栈模拟出队操作, 当出队栈空时, 把入队栈的内容弹出同时压入出队栈。

2. (1) abecfgkdhi lmj (2) abcdefghijklm



五. 算法设计:

```
1. #define maxsize 100
typedef enum{L,R} tagtype;
typedef struct
{ Bitree ptr;
  tagtype tag;
```

```

}stacknode;
typedef struct
{
    stacknode Elem[maxsize];
    int top ;
}SqStack;
Void PostOrder (Bitree t)
{
    Bitree p; SqStack s; stacknode x;
    Makenull(s);
    p=t;
    do
    {
        while (p!=null) //遍历左子树
        {
            x.ptr= p;
            x.tag=L; //标记为左子树
            push(s,x);
            p=p->lchild;
        }
        while (! Empty(s) && s.Elem[s.top].tag==R)
        {
            x=pop(s);
            p=x.ptr;
            visite(p->data); //tag 为 R，表示右子树访问完毕，故访问根节点
        }
        If (!Empty (s) && s. Elem[ s. top]. tag==R)
        {
            x=pop (s) ;
            p=x. ptr;
            visite(p->data); //tag为R，表示右子树访问完毕，
            故访问根结点
        }
        If ( ! Empty(s))
        {
            s.Elem[s.top].tag =R; //遍历右子树
            p=s.Elem[s.top].ptr->rchild;
        }
    } while ( ! Empty(s));
} PostOrder

```

```

2. int shortestpath(ALGraph *G, int i, int j)
{ // 对邻接表表示的图G，求顶点vi到顶点vj(i<>j)的最短路径
int dist[MaxVertexNum],pre[MaxVertexNum];
Queue Q; //循环数组
EdgeNode *p;
int k,t=0,w,m;

for(k=0;;k<G->n;k++)

```

```

    {dist[k]=0; pre[k]= k; }//初始化
makenull(&Q); //队列初始化
    visited[i]=TRUE;
    EnQueue(&Q, i);
    while(! Empty(&Q)){//队非空则执行
        m=DeQueue(&Q); //相当于vi出队
        p=G->adjlist[m].firstedge; //取vi的边表头指针
    wh

```

```

        mpty(s)) lem[s.top].tag =R; //遍历右子树 ! Empty(s)); 2. int
shortestpath(ALGraph *G, int i, int j) vj(i<>j)的最短路径
{
    stack
    int top; }SqStack
    void PostOrd
    { BitreMakenull(s);
    p=t; do { w { x.ptr x.tag = L; //标记为左子树 push(s,x); p=p->lchild; }
    while (
    { x = p = x.ptr; visite(p->dat 故访问根结点 } if (! E { s.E
    p=s.Elem[s.top].ptr->rchild; } }while (}//PostOrder
    { // 对邻接表表示的图G, 求顶点vi到顶点

```

