

2006 年春季学期数据结构与算法 试卷 A 参考答案

一、填空题（每空 1 分，共 12 分）

1. 根 2. 38,46,56,79,40,80 3. $O(\log n)$ $O(n \log n)$ 4. 出度 入度
5. 5、7、3、8 6. 55 7. 5 8. 链式 9. 层 10. 开放定址法、再哈希
法、链地址法、建立公共溢出区

二、单选题（每小题 1 分，共 10 分）

1. C 2. C 3. B 4. B 5. B 6. D 7. B 8. C 9. A 10. B

三、判断题（每小题 1 分，共 10 分）

- 1.(✓) 2.(✓) 3.(X) 4.(X) 5.(✓) 6.(X) 7.(✓) 8.(X) 9.(X) 10.(X)

四、简答题（每小题 4 分，共 8 分）

1. 两个栈模拟一个队列，1 号栈为入队，栈顶表示队尾；2 号栈为出队，栈顶表示队首。入队，直接进 1 号栈；出队，先判断 2 号栈是否有元素，有元素就直接弹出栈顶即队首，如果 2 号栈没有元素，则将 1 号栈的元素顺序弹出并进 2 号栈。

2.

- 0-1 : 16 (0, 1) 0-2 : 10 (0, 2) 0-3 : 14 (0, 3)
0-4 : 25 (0, 2, 4) 0-5 : 21 (0, 1, 5) 0-6 : 31 (0, 2, 4, 5)

五、算法设计（每小题 10 分，共 30 分）

1. 算法思想：可以通过层次遍历的方法来解决。不管当前结点是否有左右孩子，都入队列。这样当树为完全二叉树时，遍历时得到的是一个连续的不包含空指针的序列，反之则序列中会含有空指针。

```
typedef struct node{
    char data;
    node *lchild;
    node *rchild;
}*btree;

int JudgeCompleteTree(btree T)//判断二叉树是否为完全二叉树，是则返回1，否则返回0
{
    node *p, *q;
    makenull(Q); //清空队列
    flag = 0;
    p = T;
    EnQueue(Q, p); //建立工作队列
    while (!Empty(Q))
    {
        q = DeQueue(Q);
        if (!q)
            flag = 1;
        else if (flag)
            return 0;
        else
            EnQueue(Q, q->lchild);
            EnQueue(Q, q->rchild);
    }
    return 1;
}
```



```

    {
        EnQueue(Q, q->lchild);
        EnQueue(Q, q->rchild);
    }
}
return 1;
}
2.
typedef struct node
{
    int data;
    struct node *next;
}lklist;
void intersection(lklist *ha, lklist *hb, lklist *&hc)
{
    lklist *p, *q, *t;
    for (p = ha, hc = 0; p != 0; p = p->next)
    {
        for (q = hb; q != 0; q = q->next)
            if (q->data == p->data) break;
        if (q != 0)
        {
            t = (lklist *)malloc(sizeof(lklist));
            t->data = p->data;
            t->next = hc;
            hc = t;
        }
    }
}

```

3. 算法设计思想：1) 把边的权值按从大到小排序
 2) 取最大权边，判断是否在环路中，在则删除。
 3) 重复2) 直到边数小于顶点数

```

int C[n][n];
typedef struct
{
    int beg, end; //边的起点和终点
    int weight; //边权
}Edge;
typedef Edge E[MaxEdge];
int path(int C[n][n], int v, int w)
{ //从w顶点出发判断w到vj是否存在路径
    int i, visited[n], yes=0;
    Stack S;
    Makenull(S);
}

```


哈工大计算机考研全套视频和资料，真题、考点、典型题、命题规律独家视频讲解！

详见：网学天地（www.e-studysky.com）；咨询QQ：2696670126

```
for (i = 0; i < n; i++)
    visited[i] = 0;
Push(w, S);
visited[w] = 1;
while (!Empty(S) && yes == 0)
{
    k = Top(S);
    j = 0;
    while (C[k][j] == 1 && visited[j])
        j = j + 1;
    if (j == n)
        Pop(S);
    else if (j == v)
        yes = 1;
    else
    {
        visited[j] = 1;
        Push(j, S);
    }
}
return yes;
}
void spanning(int C[n][n], int w, int n)
{
    int i;
    Sort(E, n); //对权值按降幂排序
    i = 1;
    while (e >= n)
    {
        C[E[i].beg][E[i].end] = 0;
        C[E[i].end][E[i].beg] = 0;
        p = path();
        if (p == 1)
            e = e - 1;
        else
        {
            C[E[i].beg][E[i].end] = 1;
            C[E[i].end][E[i].beg] = 1;
        }
        i++;
    }
}
```