# 2011年真题答案

**数据结构部分**

1、

（1）只允许在表的一端进行插入，而在表的另一端进行删除，这种操作受限的线性表叫队列。循环队列是指将顺序队列假想为一个环状的空间，即把存储队列元素的表从逻辑上看成一个环。

（2）对于一个带权连通无向图G=(V,E),生成树不同，每棵树的权也可能不同。设R为G的所有生成树的集合，若T为R中边的权值之和最小的那棵生成树，则T称为G的最小生成树。构造最小生成树有普里姆算法和克鲁斯卡尔算法。构造方法见书本。

2、

（1）直接插入排序，因为直接插入排序适用于链表，而其他大多数排序只适用于顺序表。

（2）快速排序，n极大时应采用复杂度为O(log2n)的排序算法，且当排序的关键字是随机排列时，速排序的时间最短。

（3）冒泡排序，因为记录距离正确位置至多两个位置，已基本有序。

3.

算法思想：略

void LevelOrder(BTNode \*b)

{

    BTNode \*p;

BTNode \*qu[MaxSize];

int front,rear;

front=rear=-1;

    rear++;

qu[rear]=b;

while(front!=rear)

    {

        front=(front+1)%MaxSize;

        p=qu[front];

printf("%c ",p->data);

if(p->lchild!=NULL)

        {

     rear=(rear+1)%MaxSize;

qu[rear]=p->lchild;

        }

        if(p->rchild!=NULL)

        {

            rear=(rear+1)%MaxSize;

qu[rear]=p->rchild;

        }

}

}

4.

算法思想：采用头插法，遇到偶数结点就删除，然后将其插入到新的链表中。

ListNode<T>\* ptr1=L2.first; //L1游标指针  
ListNode<T>\* ptr2=L2.first; //L2游标指针  
while(L.first->link!=NULL)  
{  
 if(L.first->link->data%2!=0)  
 {  
 ptr1->link=L.first->link;  
 ptr1=ptr1->link;   
 }  
 else  
 {  
 ptr1->link=L.first->link;  
 ptr1=ptr1->link;   
 }  
 L.first->link=L.first->link->link;  
}

5.

算法思想：从右上角开始，每次将搜索值与右上角的值比较，如果大于右上角的值，则直接去除1行，否则，则去掉1列。依次类推，继续下去，算法的时间复杂度即为O(n).

bool stepWise(int mat[][], int N, int x, int &row, int &col) {

if (x< mat[0][0] || x > mat[N-1][N-1])

return false;

row = 0;

col = N-1;

while (row <= N-1 && col >= 0) {

if (mat[row][col] < target)

row++;

else if (mat[row][col] > target)

col--;

else

return true;

}

return false;

}

**操作系统部分**

6、

（1）错误，系统可以采用虚拟存储器，先将程序的一部分装入内存，而将其余部分留在外存，就可以启动程序运行，在程序执行过程中当所访问的信息不在内存时，由操作系统将所需要的部分调入内存，然后继续执行程序。通过这种方法这就可以使得300MB的程序在256MB的内存空间里运行。

（2）正确，是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。在计算机系统中，资源是有限的，而处于死锁的进程往往占有大量的资源且不能被剥夺，造成系统资源利用率降低。甚至有些死锁的出现会使系统无法正常运行，给系统造成极大危害。因此在设计操作系统时，不应该让死锁发生。

7、

（1）进程与线程的比较

1）调度：线程是独立调度的基本单位，进程是资源拥有的基本单位。在同一进程中，线程的切换不会引起进程的切换，在不同进程中进行线程切换，会引起进程切换。

2）拥有资源：线程是资源的拥有单位，而线程基本不拥有资源，但线程可以访问其隶属进程的系统资源。

3）并发性：进程之间可以并发执行，同一进程内的多个线程也可以并发执行。

4）系统开销：由于创建或撤消进程时，系统都要为之分配或回收资源，如内存空间，I/O设备等，因此操作系统所付出的开销远大于创建或撤销线程时的开销。

5）通信方面：进程间通信需要进程同步和互斥手段的辅助，以保证数据的一致性，而线程间可以直接读/写进程数据段来进行通信。

8、见书本

9、

死锁的处理策略  
1）预防死锁：设置某些限制条件，破坏产生死锁的四个必要条件中的一个或几个，以防止发生死锁。优点：适用于做突发式处理的进程，不必进行剥夺。缺点：效率低，进程初始化时间延长。

2）死锁避免：在资源的动态分配过程中，用某种方法防止系统进入不安全状态，从而避免死锁。优点：不必进行剥夺。缺点：必须知道将来的资源需求，进程不能被长时间阻塞。

3）死锁的检测与解除：无需采取任何限制性措施，允许进程在执行过程中发生死锁。通过系统的检测机构及时检测出死锁的发生，然后采取某种措施解除死锁。优点：不延长进程初始化时间，允许对死锁进行现场处理。缺点：通过剥夺解除死锁造成损失。