第一章

5、选择题：CCBDDA

6、分析时间复杂度：

（1） O(1)

（2） O(n\*m)

（3） O(n\*n)

（4） O(logn)

（5） O(n!)

（6） O()

第二章

1、选择题：BABADBCABDCCDAC

2、算法设计

1>

#include<iostream>

using namespace std;

typedef struct LNode{

    struct LNode \*next ;*//next指针*

    int data ; *//节点存储的值*

}LNode, \*LinkList;

*//初始化*

bool **Init**(LinkList &*L*){

*L* = (LNode \*)**malloc**(sizeof(LNode)) ;*//申请一个节点空间，放置头节点*

    if(*L* == **NULL**) return false ;*//内存不足，申请失败*

*L*->next = **NULL** ;

    return true ;

}

void **DisCompose**(LinkList *A*){

    LinkList B = new LNode;*//为 B申请结点空间*

    B->next = **NULL**; *// B初始化为空表*

    LinkList C = new LNode; *//为 C申请结点空间*

    C->next = **NULL**; *// C初始化为空表*

    LNode \*p;

    p = *A*->next; *// p为工作指针*

    while (p != **NULL**){

        LNode \*r;

        r = p->next; *//暂存p的后继*

        if (p->data < 0){

            p->next = B->next;

            B->next = p; *//将小于0的结点链入B表,前插法*

        }

        else{

            p->next = C->next;

            C->next = p; *//将大于等于0的结点链入C表,前插法*

            p = r; *//p指向新的待处理结点。*

        }

    }

}

2>