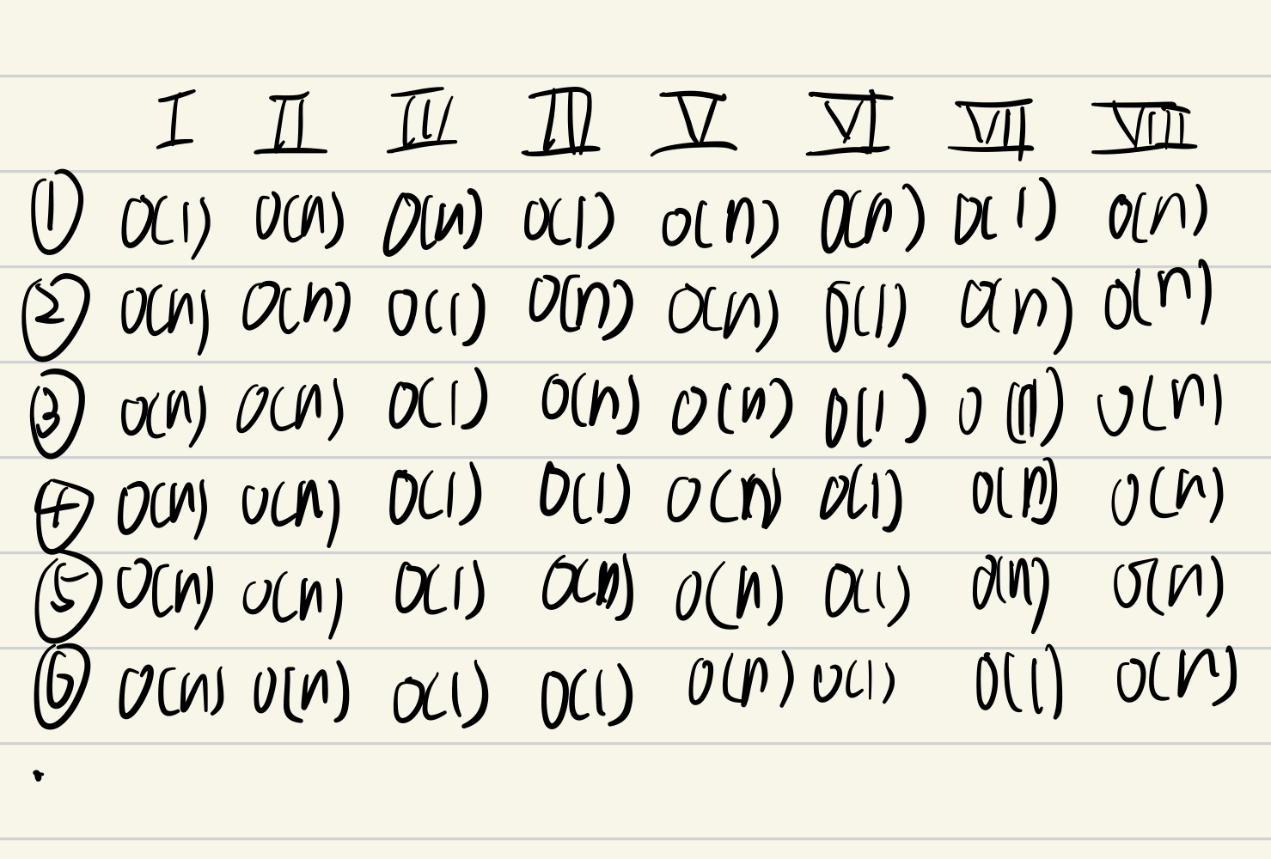
2. 简述单链表设置头结点的主要作用。

答：（1）在当单链表在其结点前删除或插入新数据的时候，只需将其头结点的指针进行修改，使修改单链表前元素变得方便。

（2）当表为空时，空表也会有头结点，使得空表和非空表的操作相同，利于操作。

3. 指出各种存储结构中对应运算算法的时间复杂度。

答：



9. 对于不带头结点的单链表L1，其结点类在为LinkNode，指出以下算法的功能。

答：将L1链表拆分成两半，使L1只含有原L1的前n/2，而L2含有原L1的后n/2。

12. 设计一个算法，将一个带头结点的数据域依次为a1、a2、…、an（n≥3）的单链表的所有结点逆置，即第一个结点的数据域变为an，第2个结点的数据域变为an-1，…，尾结点的数据域为a1。、

int fun(LinkNode\* &p) {

LinkNode\* q, \* temp;

q = p->next;

p->next = NULL;

while (q != NULL) {

temp = q->next;

q->next = p->next;

p->next = q;

q=temp;

}

}

17. 设ha=(a1，a2，…，an)和hb=(b1，b2， …，bm) 是两个带头结点的循环单链表。设计一个算法将这两个表合并为带头结点的循环单链表hc。

int fun(LinkNode\*& ha, LinkNode\*& hb, LinkNode\*& hc) {

hc = ha;

LinkNode\* p;

while (p->next != ha) {

p = p->next;

}

p->next = hb->next;

while (p->next != hb) {

p = p->next;

}

p->next = hc;

free(hb);

}