

Chapter 02

谢胡睿 2400014151 数算 (B) 24-25春

书面作业：书p67复习(4、6、7)， 算法 (3、12、16)

复习

4.

前后顺序；指针或者引用关系

6.

顺序存储：

pros: 存储密度高，且理解较为直观

cons: 删改较麻烦；需要连续空间；检索为 $O(n)$

链接存储：

pros: 插入删除灵活；空间灵活，并非静态

cons: 存取麻烦，且指针使用较为麻烦，指针内存管理较麻烦

7.

B

算法

3.

//e为顺序表

e=list()

void reverse (list &e) {

 if e==NULL:

 return

 lene=len(e)

```

for(i=0;i<lene;i++){
    temp=e[i]
    e[i]=e[lene-i-1]
    e[lene-i-1]=temp
}
return
}

```

12.

//使用一个数组储存

```

void del_repe(Llist* head ,int len){
    Llist* current{head}
    Llist* pre{head}
    flag=[ ]
    for(i=0 ;i < x; i++){
        if (current->value not in flag) //need a function to realize the "in"
            flag.add(current->value)
        else {
            if (current->value in flag)
                Llist* temp{nullptr}
                pre->next=current->next
                free(current)
                current=pre->next
            else{
                pre=current
                current=current->next
            }
        }
    }
}

```

16.

```

Llist* combine(Llist* a,Llist* b){
    Llist* chead=new Llist{nullptr}
    Llist* c=chead->next
    List* acurrent{a->head} ,bcurrent{b->head}
    if(acurrent->next== nullptr && bcurrent->next== nullptr){
        return c
    }//robustness
}

```

```

while(acurrent->next!= nullptr || bcurrent->next!= nullptr){//判空
    if (acurrent->next!= nullptr && bcurrent->next!= nullptr){
        c->value=acurrent->value>bcurrent->value?acurrent->value:bcurrent->value
        bcurrent=bcurrent->next
        acurrent=acurrent->next
    }
    else{
        if(acurrent->next==nullptr) {//尾巴
            c->value=bcurrent->value
            bcurrent=bcurrent->next
        }

        if(bcurrent->next==nullptr) {//尾巴
            c->value=acurrent->value
            acurrent=acurrent->next
        }
    }
    c=c->next
}
return chead
}

```