实验一 路凯 201930310034

void Getmaxin(S qList \*L)

{

int i = 0, MAX, MIN;

MAX = (\*L).elem[i];

MIN = (\*L).elem[i];

while (i<(\*L).length && (\*L).elem[i])

{

if ((\*L).elem[i] >= MAX)

MAX = (\*L).elem[i];

if ((\*L).elem[i] <= MIN)

MIN = (\*L).elem[i];

i++;

}

printf("MAX=%d\nMIN=%d\n", MAX, MIN);

}

左移函数

Status CycleLeft(SqList \*L, int p) //循环左移

{

int i;

SqList S;

S.elem = (ElemType\*)malloc(LIST\_INIT\_SIZE\*sizeof(ElemType));

if (!(S).elem)

exit(OVERFLOW);

if (p<0 || p>(\*L).length)

return false;

for (i = 0; i < p; i++)

{

S.elem[i] = (\*L).elem[i];

}

for (i = p; i < (\*L).length; i++)

{

(\*L).elem[i-p] =(\*L).elem[i];

}

for (i = 0; i < p; i++)

{

(\*L).elem[(\*L).length-p+i]=S.elem[i];

}

DestroyList(&S);

return OK;

}

主函数

void main()

{

SqList L1, L2, L3;

//int e;

int i;

if (InitList(&L1) == 1 && InitList(&L2) == 1)

{

printf("顺序表初始化成功\n");

}

for (i = 1; i<10; i++)

ListInsert(&L1, i, 2 \* i);

for (i = 1; i<10; i++)

ListInsert(&L2, i, (2 \* i + 1));

MergeList(L1, L2, &L3);

Getmaxin(&L3);

CycleLeft(&L3,10);

ListTraverse(L3);

DestroyList(&L1);

DestroyList(&L2);

DestroyList(&L3);

}

总结：线性表顺序存储结构存在一定缺点，即插入删除需要移动大量元素。

顺序存储结构更适合于查找和存取，链式存储结构更适用于插入和删除、修改等。