一、List 链表容器

链表是一种物理存储单元上不连续的存储结构，它是通过指针链接起来的。一个结点包括两个部分，一部分是存储数据的，一部分是存储下一个结点地址的指针。

List链表容器扩充空间是需要多少给多少，插入一个就多链接一个结点，删除一个就释放一个结点。

List容器是一个循环的双向链表，List链表容器对于元素的插入和删除，都是常数时间复杂度的，但是，对于查找元素，时间复杂度就非常高。

二．优缺点：

1. 动态存储，不会浪费内存
2. 插入删除方便，但查找费时

三．List容器的迭代

由于list是一个循环的双向链表，迭代器必须能够具备前移、后移的能力，因此迭代器是双向迭代器。同时由于链表是不连续存储的，不支持随机访问，不可以用系统提供的算法，但是类的内部会提供。

如： sort算法不支持list容器，但是list容器自带着sort成员函数。

List<int> l;

l.sort();

不连续存储的链表另一个特点是：插入和删除操作不会影响原有的list迭代器失效。因为各个结点都可以说是独立的。

四．List容器的常用操作

（一）list构造函数

list<T> lstT;//list采用采用模板类实现,对象的默认构造形式：

list(beg,end);//构造函数将[beg, end)区间中的元素拷贝给本身。

list(n,elem);//构造函数将n个elem拷贝给本身。

list(const list &lst);//拷贝构造函数。

（二）list元素插入和删除操作

push\_back(elem);//在容器尾部加入一个元素

pop\_back();//删除容器中最后一个元素

push\_front(elem);//在容器开头插入一个元素

pop\_front();//从容器开头移除第一个元素

insert(pos,elem);//在pos位置插elem元素的拷贝，返回新数据的位置。

insert(pos,n,elem);//在pos位置插入n个elem数据，无返回值。

insert(pos,beg,end);//在pos位置插入[beg,end)区间的数据，无返回值。

clear();//移除容器的所有数据

erase(beg,end);//删除[beg,end)区间的数据，返回下一个数据的位置。

erase(pos);//删除pos位置的数据，返回下一个数据的位置。

remove(elem);//删除容器中所有与elem值匹配的元素。

当用list容器的remove操作删除自定义的数据类型时，根据list容器的remove源码可以得知。需要重载 == 操作符。

（三）list容器大小操作

size();//返回容器中元素的个数

empty();//判断容器是否为空

resize(num);//重新指定容器的长度为num，

若容器变长，则以默认值填充新位置。

如果容器变短，则末尾超出容器长度的元素被删除。

resize(num, elem);//重新指定容器的长度为num，

若容器变长，则以elem值填充新位置。

如果容器变短，则末尾超出容器长度的元素被删除。

（四）list容器赋值操作

assign(beg, end);//将[beg, end)区间中的数据拷贝赋值给本身。

assign(n, elem);//将n个elem拷贝赋值给本身。

list& operator=(const list &lst);//重载等号操作符

swap(lst);//将lst与本身的元素互换。

（五）list容器元素的存取

front();//返回第一个元素。

back();//返回最后一个元素。

（六）。List反转排序

reverse();//反转链表，比如lst包含1,3,5元素，运行此方法后，lst就包含5,3,1元素。

sort(); //list排序