Deque容器

1. Deque容器的概念

之前我们介绍了vector容器的概念，vector容器相当于数组，不过vector的空间扩充是动态的。而且vector容器是单向开口的，虽然也允许在头部插入元素，但是效率非常低。Deque容器是双向开口的连续线性空间，可以在头尾两端进行元素的插入和删除操作。

Deque容器使用常量时间复杂度在头部进行元素的插入和删除操作。

Deque容器是以动态的分段连读空间链接而成，需要扩充空间时，就链接一段新的空间。

1. Deque容器的实现原理

Deque容器是由一些连续空间链接而成的，当我们需要为deque容器扩充时，则配置一段连续空间，链接在Deque的头部或者尾部。这样，就避免了vector容器的配置空间，复制内容，释放原空间的操作。但是，这也是有代价的，deque容器需要维护这些分段的内存空间，并提供随机存取的接口，因此，deque容器的迭代器结构十分复杂。

Deque容器划出一小块内存空间（中控器）存储着各个分段内存空间的地址，这些分段内存空间称为缓冲区。这些缓冲区中即存储着deque容器的内容。

1. Deque容器的函数操作
2. deque构造函数

deque<T> deqT;//默认构造形式

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

deque(n, elem);//构造函数将n个elem拷贝给本身。

deque(const deque &deq);//拷贝构造函数。

1. deque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。

deque& operator=(const deque &deq); //重载等号操作符

swap(deq);// 将deq与本身的元素互换

1. deque大小操作

deque.size();//返回容器中元素的个数

deque.empty();//判断容器是否为空

deque.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。

deque.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。

1. deque两端插入和删除操作

push\_back(elem);//在容器尾部添加一个数据

push\_front(elem);//在容器头部插入一个数据

pop\_back();//删除容器最后一个数据

pop\_front();//删除容器第一个数据

1. deque数据存取

at(idx);//返回索引idx所指的数据，如果idx越界，抛出out\_of\_range。

operator[];//返回索引idx所指的数据，如果idx越界，不抛出异常，直接出错。

front();//返回第一个数据。

back();//返回最后一个数据

1. deque插入操作

insert(pos,elem);//在pos位置插入一个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。

1. deque删除操作

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。