**9 顺序容器**

**1，顺序容器类型**

vector：可变大小数组。支持快速随机访问。在尾部之外的位置插入或删除元素可能很慢。

deque：双端队列，支持快速随机访问，在头尾位置插入删除速度很快。

list：双向链表，只支持双向顺序访问。在list中任何位置进行插入删除操作速度都很快。

forward\_list：单向链表，只支持单向顺序访问。在链表任何位置进行插入删除操作速度都很快。

array：固定大小数组，支持快速随机访问，不能添加或删除元素。

string：与vector相似的容器，但专门用于保存字符。随机访问快。在尾部插入/删除速度快。

通常，使用vector是最好的选择，除非你有很好的理由选择其他容器。

**2，容器基本操作**

赋值与swap

c = {a,b,c,d}

a.swap(b)等价于swap(a,b)

大小

c.size() c.max\_size() c.empty()

获取迭代器

c.begin(),c.end()

c.cbegin(),c.cend()

**3，顺序容器操作**

向顺序容器添加元素的操作

c.push\_back(t)

c.emplace\_back(args)

c.push\_front(t)

c.emplace\_front(args)

c.insert(p,t)在迭代器p指向的元素之前创建t元素。返回指向新增元素的迭代器。

c.emplace(p,args)

c.insert(p,n,t)插入n个t。

c.insert(p,b,e)

顺序容器中访问元素的操作（返回的都是引用）

c.back()

c.front()

c[n]

c.at(n)

顺序容器的删除操作

c.pop\_back()

c.pop\_front()

c.erase(p)

c.erase(b,e)

c.clear()

**4，容器适配器**

除了顺序容器外，标准库还定义了三个顺序容器适配器：stack,queue和priority\_queue。

默认情况下，stack和queue都是基于deque实现的，priority\_queue实在vector之上实现的。

参考文档：栈的操作

参考文档：queue和priority\_queue的操作