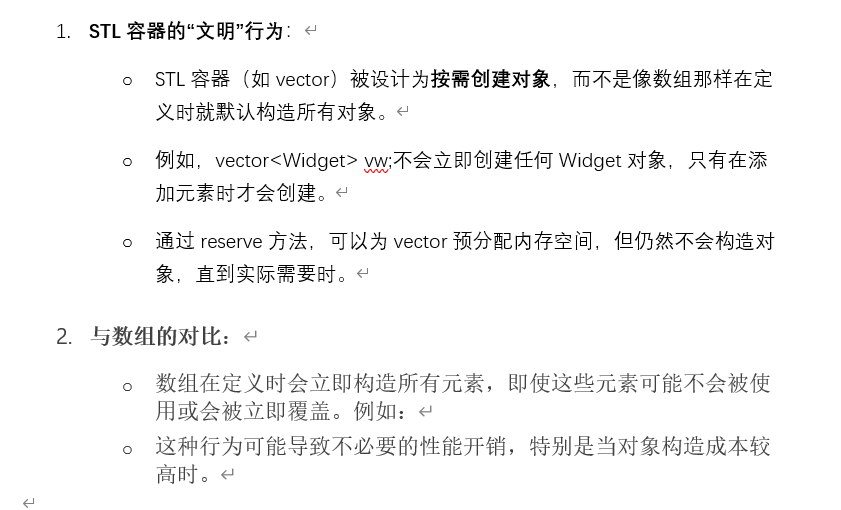
序列容器



如果你要将新创建的对象添加到向量中，可以利用 emplace\_back() 函数，它会直接在向量中创建对象，而不是先创建对象，再使用 push\_back() 函数将其复制或移动到向量中

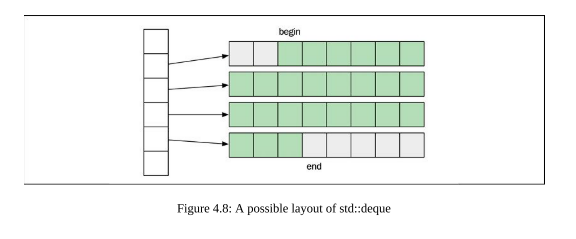
调用empty而不是检查size（）是否为零。

容器消除会自动删除对象，但是指针不会。当使用指针并且new创建对象时，记得删除对象，用智能指针。

当发生插入操作导致容量不足时，会重新分配内存，将原有的元素复制或移动到新的内存位置。如果元素类型有 noexcept 移动构造函数，就会优先使用移动构造，否则使用复制构造。

从 C++20 开始，还有两个自由函数可以从 std::vector 中删除元素。在 C++20 之前，我们必须使用 “erase - remove” 惯用法，我们将在第 5 章 “算法” 中讨论这个问题。然而，现在从 std::vector 中删除元素的推荐方法是使用 std::erase() 和 std::erase\_if()。

在 C++17 之前，std::basic\_string 在内存中的布局不一定是连续的，这就限制了它直接作为字符数组传递给某些需要连续内存的 API。但从 C++17 开始，std::basic\_string 保证了其内部字符存储是连续的，这样就可以通过取第一个元素的地址（如 &content[0]）将其当作字符数组来使用。



有序关联容器：这些容器基于树;容器使用树来存储它们的元素。O(log n)

无需关联容器哈希表，遇到哈希冲撞变为链表。O(1)

在本节中，我们将讨论 C++ 标准库中一些相对较新的类模板：C++17 引入的 std::string\_view 以及 C++20 引入的 std::span

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。