序列：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序列 | 名称 | 介绍 | 随机访问 | 特点 | 使用场景 |
| 元素按照严格的线性顺序排列，存在第一个元素、最后一个元素，除第一个元素和最后一个元素外，每个元素前后都分别有一个元素 | vector | 一种动态数组，顺序排列元素 | 支持 | 自动管理内存，对象长度动态改变，可反转，最简单的序列 | 需要快速查找，不需频繁插入和删除，除特殊需要外默认使用 |
| deque | 双端队列 | 支持 | 可以在队列的两端进行元素的插入和删除操作 | 多数操作发生在序列的起始和结尾处 |
| list | 双向链表，除第一个和最后一个元素外，每个元素都与前后的元素相链接 | 不支持 | 可反转，容器插入或删除元素后，链表迭代器指向元素不变 | 需要频繁插入和删除，不需要快速查找 |
| queue | 一个适配器类，让底层类（默认为deque)展示典型的队列接口 | 不允许 | 不允许遍历队列，可以将元素添加到队尾、从队首删除元素、查看队首和队尾的值、检查元素数目和测试队列是否为空 | 先进先出（FIFO） |
| stack | 一个适配器类，给底层类（默认vector）提供典型的栈接口 | 不允许 | 不允许遍历栈，可以将压入元素推到栈顶、从栈顶弹出元素、查看栈顶的值、检查元素数目和测试栈是否为空 | 先进后出（FILO） |
| array | 大小固定的数组 | 允许 | 不支持添加或删除元素等改变大小的操作，定义时不仅要指定元素类型，还要指定容器大小 | 已知元素数量 |

关联容器：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 关联容器 | 名称 | 键的唯一性 | 特点 | 使用场景 |
| 将值与键关联在一起，并使用键来查找值 | set | 唯一 | 值类型与键类型相同，每个键对应一个值，可反转，可排序 | 需要元素有序，查找、删除和插入性能一样 |
| map | 值类型与键类型不同，每个键对应一个值 | 需要键有序将值关联，查找、删除和插入性能一样 |