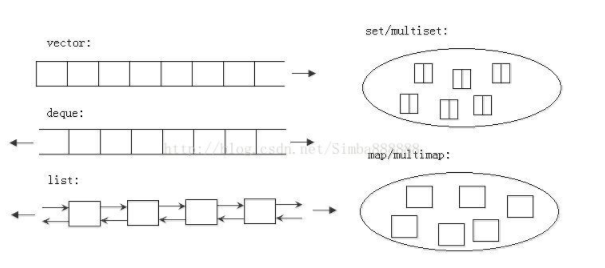
STL是由哪些组件构成的？它们各自具备什么样的特点？

六大组件：

1. **容器(Container)：**



1、需要频繁在序列中间位置上进行插入和/或删除操作且不需要过多地在序列内部进行长距离跳转，应该选择list

2、vector头部与中间插入删除效率较低，在尾部插入与删除效率高。

3、deque是在头部与尾部插入与删除效率较高

序列式容器: 向量（vector）、双端队列（deque）、列表（list）

1. 序列式容器Sequence containers,其中每个元素均有固定位置——取决于插入时机和地点，和元素值无关

关联式容器: 集合（set）、多重集合（multiset）、映射（map）和多重映射（multimap）

1. 关联式容器Associative containers，元素位置取决于特定的排序准则以及元素值，和插入次序无关

**2．迭代器(Iterator)：**

1、迭代器Iterators，用来在一个对象群集（collection of objects）的元素上进行遍历。这个对象群集或许是个容器，或许是容器的一部分。迭代器的主要好处是，为所有容器提供了一组很小的公共接口。迭代器以++进行累进，以\*进行提领，因而它类似于指针，我们可以把它视为一种smart pointer

2、比如++操作可以遍历至群集内的下一个元素。至于如何做到，取决于容器内部的数据组织形式。

3、每种容器都提供了自己的迭代器，而这些迭代器能够了解容器内部的数据结构。

**3．适配器(Adapter):**

适配器是一种接口类，为已有的类提供新的接口，目的是简化、约束、使之安全、隐藏或者改变被修改类提供的服务集合

1. 容器适配器：用来扩展7种基本容器，它们和顺序容器相结合构成栈、队列和优先队列容器
2. 迭代器适配器：反向迭代器、插入迭代器、IO流迭代器
3. 函数适配器：函数对象适配器、成员函数适配器、普通函数适配器

**4. 算法库(Algorithm)**

算法Algorithms，用来处理群集内的元素。它们可以出于不同的目的而搜寻、排序、修改、使用那些元素。通过迭代器的协助，我们可以只需编写一次算法，就可以将它应用于任意容器，这是因为所有的容器迭代器都提供一致的接口。

**5. 函数对象(function object)**

1、函数对象（function object）也称为仿函数（functor）

2、一个行为类似函数的对象，它可以没有参数，也可以带有若干参数。

3、任何重载了调用运算符operator()的类的对象都满足函数对象的特征

4、函数对象可以把它称之为smart function。

5、STL中也定义了一些标准的函数对象，如果以功能划分，可以分为算术运算、关系运算、逻辑运算三大类。为了调用这些标准函数对象，需要包含头文件<functional>。

**6. 空间配置器（allocator）**

负责空间配置与管理。从实现的角度来看，配置器是一个实现了动态空间配置、空间管理、空间释放的class template。