**Vector小结**

vector是STL中最常见的容器，它是一种顺序容器，支持随机访问。vector是一块**连续分配**的内存，从数据安排的角度来讲，和数组极其相似，不同的地方就是：数组是静态分配空间，一旦分配了空间的大小，就不可再改变了；而vector是**动态分配空间**，随着元素的不断插入，它会按照自身的一套机制不断扩充自身的容量。

vector的**扩充机制**：按照容器现在容量的一倍进行增长。vector容器分配的是一块连续的内存空间，每次容器的增长，并不是在原有连续的内存空间后再进行简单的叠加，而是重新申请一块更大的新内存，并把现有容器中的元素逐个复制过去，然后销毁旧的内存。这时原有指向旧内存空间的**迭代器**已经失效，所以当操作容器时，迭代器要及时更新。

//迭代器是随机存取迭代器，对vector来说就是指针，迭代器持续有效

//除非在一个较小索引位置插入删除元素或者内存重新分配

vector的元素是可以任意类型的T，但必须具备赋值和拷贝两个性质。

在末端添加和删除元素，vector性能很好，但是在前端中端很差，因为当前操作点之后的每一个元素都必须移到另一个位置，每一次都会调用赋值操作。

vector类常用的函数如下所示：

    **1.构造函数**

vector():创建一个空vector

vector(int nSize):创建一个vector,元素个数为nSize

vector(int nSize,const t& t):创建一个vector，元素个数为nSize,且值均为t

vector(const vector&):复制构造函数

vector(begin,end):复制[begin,end)区间内另一个数组的元素到vector中

    **2.增加函数**

void push\_back(const T& x):向量尾部增加一个元素X

iterator insert(iterator it,const T& x):向量中迭代器指向元素前增加一个元素x

iterator insert(iterator it,int n,const T& x):向量中迭代器指向元素前增加n个相同的元素x

iterator insert(iterator it,const\_iterator first,const\_iterator last):向量中迭代器指向元素前插入另一个相同类型向量的[first,last)间的数据

   **3.删除函数**

iterator erase(iterator it):删除向量中迭代器指向元素

iterator erase(iterator first,iterator last):删除向量中[first,last)中元素

void pop\_back():删除向量中最后一个元素

void clear():清空向量中所有元素

  **4.遍历函数**

reference at(int pos):返回pos位置元素的引用

reference front():返回首元素的引用

reference back():返回尾元素的引用

iterator begin():返回向量头指针，指向第一个元素

iterator end():返回向量尾指针，指向向量最后一个元素的下一个位置

reverse\_iterator rbegin():反向迭代器，指向最后一个元素

reverse\_iterator rend():反向迭代器，指向第一个元素之前的位置

  **5.判断函数**

bool empty() const:判断向量是否为空，若为空，则向量中无元素

  **6.大小函数**

int size() const:返回向量中元素的个数

int capacity() const:返回当前向量张红所能容纳的最大元素值

int max\_size() const:返回最大可允许的vector元素数量值

  **7.其他函数**

void swap(vector&):交换两个同类型向量的数据

void assign(int n,const T& x):设置向量中第n个元素的值为x

void assign(const\_iterator first,const\_iterator last):向量中[first,last)中元素设置成当前向量元素

参考：

[1][vector 基本用法详解](http://www.cnblogs.com/panjiangfy/archive/2012/05/30/vector.html)

http://www.cnblogs.com/panjiangfy/archive/2012/05/30/vector.html

[2]STL之二：vector容器用法详解

http://blog.csdn.net/longshengguoji/article/details/8507394

[3][C++ vector用法](http://www.cnblogs.com/wang7/archive/2012/04/27/2474138.html)

http://www.cnblogs.com/wang7/archive/2012/04/27/2474138.html