查找算法

一．Find算法

查找基本数据类型元素

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param value 查找的元素

@return 返回查找元素的位置

Void test()

{

Vector<int> v;

For(int I = 0; I < 10; ++i)

{  
v.push\_back(i);

}

Vector<int>::iterator pos = find(v.begin(), v.end(), 5);

If( pos != v.end())

{  
cout << “找到了数据” << \*pos << endl;

}

Else

Cout <<”未找到!” <<endl;

}

查找自定义元素

需要重载 ==运算符

Class Person  
{

Public:

Person(string name, int age)

{  
this->m\_Name = name;

This->m\_Age = age;

}

Bool operator==(const Person&p)

{  
if(this->m\_Name == p.m\_Name && this->m\_Age == p.m\_Age)

Return true;

Return false;

}

String m\_Name;

String m\_Age;

}

Void test()

{  
vector<Person> v;

Person p1(“a”, 1);

Person p2(“b”, 2);

v.push\_back(p1);

v.push\_back(p2);

vector<Person>::iterator pos = find(v.begin(), v.end(), p2);  
}

find\_if条件查找

通过回调函数或者谓词（返回bool类型的函数对象）

find\_if算法 条件查找

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param callback 回调函数或者谓词(返回bool类型的函数对象)

@return bool 查找返回true 否则false

Class Person

{  
public:

Person(string name, int age)

{  
this->m\_Name = name;

This->m\_Age = age;

}

String m\_Name;

Int m\_Age;

}

Class myCompare

{  
public:

Bool operator()(Person \* p1)

{

If(p1->m\_Name == “b” && p1->m\_Age == 2)

Return true;

Return false;

}

}

Void Test()

{  
vector<Person\*>v;

Person p1(“a”,1);

Person p2(“b”, 2);

v.push\_back(&p1);

v.push\_back(&p2);

Person \*p = new Person(“b”, 2);

Vector<Person\*>::iterator pos = Find\_if(v.begin(), v.end(), myCompare());

}

Adjacent\_find算法

查找相邻重复元素，返回相邻元素的第一个位置的迭代器

adjacent\_find算法 查找相邻重复元素

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param \_callback 回调函数或者谓词(返回bool类型的函数对象)

@return 返回相邻元素的第一个位置的迭代器

比较简单，不展示了…

Binary\_search算法

二分法查找，返回bool类型，查找是否成功。

binary\_search算法 二分查找法

注意: 在无序序列中不可用

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param value 查找的元素

@return bool 查找返回true 否则false

Count算法

统计元素出现次数

count算法 统计元素出现次数

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param value回调函数或者谓词(返回bool类型的函数对象)

@return int返回元素个数

Count\_if算法

同样是统计元素出现次数

配有回调函数或者谓词

count算法 统计元素出现次数

@param beg 容器开始迭代器

@param end 容器结束迭代器

@param callback 回调函数或者谓词(返回bool类型的函数对象)

@return int返回元素个数