



(!\_Pred(\_Left, \_Right)) return (false ). else if (\_Pred(\_Right, \_Left))  
\_DEBUG\_ERROR2("invalid operatorlt. , 编译器中看到 \_Pred 的  
value 是 {lessthan} , type 是 std::less gt. , 但这里有更大的发现  
: strict weak ordering!!! 自己 C 功底很浅 , 这是一个新的发现  
, 马上 google 一下 "strict weak ordering" 这个关键词 , 果然  
发现大量的专题链接 ! 暂且先放下这个主题。问题猜测肯定是  
出在 operator lt. 定义 : {31, 41} lt. {31, 41} 的返回值也是 false  
, 那么 , 由这两个比较能得出结论 : {31, 41} == {31, 59} !!! 这  
也无怪乎程序运行会返回不期望的结果了。但这也只是猜测  
, 继续调试 , 看能不能找到 \_Pred 函数的真实面目。上面说了  
从编译器中看出 \_Pred 的 type 是 std::less gt. , 在 MSDN 中找到  
less 是 STL 中的一个模板类 , 以下是在 MSDN 中看到的定义 :  
less less templategt. struct less : public binary\_function gt. {  
bool operator() (const T& x, const T& y) const. }. The template class defines  
its member function as returning x lt. y. class \_Ty { \_Ty, bool lt.  
bool operator ()(const \_Ty& \_Right) const { // apply operatorlt.  
\_Right). } }. 它最终比较 \_Left 和 \_Right 时调用的正是 struct S 中  
定义的 operator lt. "。这个问题就不再重现调试了 , 它的解  
决方法见下列地址 : <http://support.microsoft.com/kb/949171>  
strict weak ordering 是一个数学上的术语 , 刚刚给出的这个地  
址上面有关于 strict weak ordering 的简明的解释 , 贴过来 :  
The STL algorithms for stable\_sort ( ) and sort() require the binary  
predicate to be strict weak ordering. For example: Strict: pred (X, X)  
is always false. Weak: If ! pred (X, Y) amp. !pred (Y, X), X==Y.  
Ordering: If pred (X, Y) amp. pred (Y, Z), then pred (X, Z). 100Test  
下载频道开通 , 各类考试题目直接下载。详细请访问

