
Subject: Change in <Core/Sort.h> breaks my code
Posted by [Lance](#) on Sun, 08 Nov 2020 18:12:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

I recently updated to 20111108 from a version that's about half year old, and encountered a compilation error. As I ponder about the error, it appears the new implementation of Sort__ in <Core/Sort.h> requires the Iterator class to supply a <= operator. While I don't know how the Iterator specification says, I believe it's probably better to revise the Sort__ code slightly to relieve the library user of the obligation to supply a <=, however trial it is.

```
void Sort__(l l, l h, const Less& less)
{
    int count = int(h - l);
    l middle = l + (count >> 1);    // get the middle element
    for(;;) {
        switch(count) {
            case 0:
            case 1: return;
            case 2: OrderIter2__(l, l + 1, less); return;
            case 3: OrderIter3__(l, l + 1, l + 2, less); return;
            case 4: OrderIter4__(l, l + 1, l + 2, l + 3, less); return;
            case 5: OrderIter5__(l, l + 1, l + 2, l + 3, l + 4, less); return;
            case 6: OrderIter6__(l, l + 1, l + 2, l + 3, l + 4, l + 5, less); return;
            case 7: OrderIter7__(l, l + 1, l + 2, l + 3, l + 4, l + 5, l + 6, less); return;
        }
        if(count > 1000) {
            middle = l + (count >> 1); // iterators cannot point to the same object!
            l q = l + 1 + (int)Random((count >> 1) - 2);
            l w = middle + 1 + (int)Random((count >> 1) - 2);
            OrderIter5__(l, q, middle, w, h - 1, less);
        }
        else
            OrderIter3__(l, middle, h - 1, less);

        l pivot = h - 2;
        IterSwap(pivot, middle); // move median pivot to h - 2
        l i = l;
        l j = h - 2; // l, h - 2, h - 1 already sorted above
        for(;;) { // Hoare's partition (modified):
            while(less(*++i, *pivot));
            do
                if(j <= i) goto done;
            *****
            *****SHOULD BE*****
            if(!(i<j) ) goto done;
            *****

            while(!less(*--j, *pivot));
```

```
    IterSwap(i, j);  
}  
done:
```

Thank you!
