

nth_element 指定元素排序

1- 模板函数

头文件: `<algorithm>`

函数名: `std::nth_element`

对给定范围[first,last)内的元素进行重新布置.方法是,nth位置的元素放置的值就是把所有元素排序后在nth位置的值.把所有不大于nth的值放到nth的前面,把所有不小于nth的值放到nth后面.对**给定范围内的元素"排序"**.【有点像快排】

函数原型

```
template <class RandomAccessIterator>
void nth_element(RandomAccessIterator first, RandomAccessIterator nth, RandomAccessIterator last);

template <class RandomAccessIterator, class StrictWeakOrdering>
void nth_element(RandomAccessIterator first, RandomAccessIterator nth, RandomAccessIterator last,
StrictWeakOrdering comp);    // 最后一个变成了严格弱排序？
```

2- 参数

- first, last 随机访问迭代器.指定了需要重新"排序"的范围.包括first,但不包括last.
 - nth 随机访问迭代器.指向范围[first,last)内的一个位置.这个位置将放置排序后应该放于此位置的元素.
 - comp二元函数. 返回bool. 表明是否第一个参数应该排序到第二个参数的前面.此函数不应该修改参数值.可以是一个函数指针或函数对象.

3- 返回

void

简单的说nth_element算法仅排序第nth个元素（从0开始的索引）

如iarray [first,last) 元素区间排序后 iarray[nth] 就是第nth大的元素（从0开始）

要注意的是[first,nth) [nth,last)内 的大小循序还不一定，只能确定iarray[nth]是第nth大的元素。

当然 [first,nth) 肯定是不大于 [nth,last)的。

4-简单测试代码如下

要注意的是，此函数只是将第nth大的元素排好了位置，但并没有返回值

所以要知道第nth大的元素 还得进行一步，`cout<<iarray[nth]<<endl;` nth既那个位子

```
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    int iarray[]={5,6,15,89,7,2,1,3,52,63,12,64,47};
    int len=sizeof(iarray)/sizeof(int);
    int i;
    for(i=0;i<len;i++)
        cout<<iarray[i]<<" ";
    nth_element(iarray,iarray+6,iarray+len);    //排序第6个元素
    cout<<endl;

    for(i=0;i<len;i++)
        cout<<iarray[i]<<" ";
    cout<<" 第6-th个元素    "<<iarray[6]<<endl;
}
```

5- 另一个例子

班上有10个学生，我想知道分数排在倒数第4名的学生。

如果要满足上述需求，可以用sort排好序，然后取第4位（因为是由小到大排），更聪明的朋友会用partial_sort，只排前4位，然后得到第4位。其实这时你还是浪费，因为前两位你根本没有必要排序，此时，你就需要nth_element:

对于上述实例需求，你只需要按下面要求修改1.4中的程序：???

```
stable_sort(vect.begin(), vect.end(),less<student>());
```

替换为：

```
nth_element(vect.begin(), vect.begin()+3, vect.end(),less<student>());
```

运行结果为：

```
-----before sort...
Tom: 74
Jimmy: 56
Mary: 92
Jessy: 85

Jones: 56
```

```
Bush: 52
Winter: 77
Andyer: 63
Lily: 76
Maryia: 89
-----after sort ....
Jone: 56
Bush: 52
Jimmy: 56
Andyer: 63
Jessy: 85
Mary: 92
Winter: 77
Tom: 74
Lily: 76
Maryia: 89
```

第四个是谁？Andyer，这个倒霉的家伙。为什么是begin()+3而不是+4？我开始写这篇文章的时候也没有在意，后来在ilovevc 的提醒下，发现了这个问题。begin()是第一个，begin()+1是第二个，... begin()+3当然就是第四个了。