<https://www.cnblogs.com/AndyJee/p/4676146.html>

**qsort的用法**

原型：void qsort(void \*base, int nelem, int width, int (\*fcmp)(const void \*, const void \*))

参数：

Base:待排序数组首地址

nelem：数组中待排序元素数量

width：各元素的占用空间大小

fcmp：指向函数的指针，用于确定排序的顺序

对于vector也同样适应：

int a[] = { 2, 7, 11, 15 };

vector<int> nums;

for (int i = 0, size = sizeof(a) / sizeof(int); i < size; i++) nums.push\_back(a[i]);

cout << "size = " << nums.size() << endl;

qsort(&nums.at(0), nums.size(), sizeof(int), [] (const void \*a, const void \*b) -> int {

return \*(int \*)b - \*(int \*)a;

});

for (vector<int>::iterator itor = nums.begin(); itor != nums.end(); itor++)

{

cout << \*itor << " ";

}



**sort的用法**

sort有默认的比较函数，默认是按升序排列

void sort(RanIt first, RanIt last, Pred pr)前两个参数为迭代器指针，分别指向容器的首尾，第三个参数为比较方法

int a[] = { 2, 11, 7, 15 };

vector<int> nums;

for (int i = 0, size = sizeof(a) / sizeof(int); i < size; i++) nums.push\_back(a[i]);

sort(nums.begin(), nums.end(), [] (int a, int b) -> bool {

return a < b;

});

for (vector<int>::iterator itor = nums.begin(); itor != nums.end(); itor++) cout << \*itor << " ";



**qsort与sort的区别**

sort是一个改进版的qsort

sort函数优于qsort的一些特点：对大数组采取9项取样，更完全的三路划分算法，更细致的对不同数组大小采用不同方法排序。