快速排序

需要注意的事项是：
1- partition 和 quickSort 都要传入 start 和 end 索引
2- partition 返回的是small 的坐标
3- 为了避免自己和自己交换，可以加一步判断
4- 时间复杂度： O(nlgn)，最差是O(n^2)。空间复杂度是需要O(nlgn)的辅助内存【交换过程中产生的辅助内存】。 不稳定

快速排序代码如下：

```
#include<iostream>
#include<stdlib.h>
#include<time.h>
#include <vector>

using namespace std;
int partition(vector<int>& vec,int start, int end){
    /// 边界检查在这里
    if(vec.size() == 0 || start < 0 || end >= vec.size())
        throw ;

    srand(time(NULL));
    int index = rand() %( end - start + 1 ) + start;  /// 产生[0,vec.size()-1]随机数
    swap(vec[index],vec[end]);
    int small = start -1;  // 指向当前最小的元素
    for( int i = start; i < end; i++ ){
        if( vec[i] < vec[end]  ) {
            small ++;
            /// 为了防止自身与自身交换，可以加一步
            if( i != small )
                swap( vec[i], vec[small] );
        }
    }

    /// 交换最后一个
    small ++;
    swap(vec[small],vec[end]);
    return small;                           /// ！！！ 注意这里返回的是small，不是index
}
```

```
void quickSort(vector<int>& vec, int start, int end){
    if( start == end ) return;

    int index = partition(vec, start, end); /// 注意这里一定是start 和 end
    if( index > start )
        quickSort(vec,start, index -1);

    if( index < end )
        quickSort(vec,index+1, end);
}

int main()
{
    vector<int> in {1,3,2,5,8,3,9 ,8,7,11};
    quickSort(in,0,in.size()-1);
    for( int val: in )
        cout << val << " ";

    return 0;
}
```