

## 快速

```
<?php
$arr=array(1,3,5,7,9);
print_r(quick_sort($arr));
// 7 为基数 比基数小的放左边数组 否则放右边数组
// 第一次 5<7 left[5]
// 第二次 4<7 left[5,4]
// 第三次 2<7 left[5,4,2]
// 第四次 50!<7 right[50]
// 第五次 100!<7 right[50,100]
// 第六次 70!<7 right[50,100,70]
// 第七次 30!<7 right[50,100,70,30]
// 完成一趟排序: 5,4,2,7,50,100,70,30
// 利用递归 把左右的数组各自排序 然后合并数组
// 例如: 左边数组
// 第一次 4<5 left[4]
// 第二次 4<5 left[4,2]
// 继续拿4为基数做比较
// 第一次 2<4 left[2]
// 又完成一趟排序 为 2,4,5
// 依次完成快速排序
//
function quick_sort($arr) {
    //先判断是否需要继续进行
    $length = count($arr);
    if($length <= 1) {
        return $arr;
    }
    //选择第一个元素作为基准
    $base_num = $arr[0];
    //遍历除了标尺外的所有元素,按照大小关系放入两个数组内
    //初始化两个数组
    $left_array = array(); //小于基准的
    $right_array = array(); //大于基准的
    $c=0;
    for($i=1; $i<$length; $i++) {
        if($base_num > $arr[$i]) {
            //放入左边数组
            $left_array[] = $arr[$i];
            $c++;
        }
        // print_r($left_array);
        // else {
        //     $c++;
        //     //放入右边
        //     $right_array[] = $arr[$i];
        // }
    }
    echo $c;die;
    // print_r($right_array);die;
    //再分别对左边和右边的数组进行相同的排序处理方式递归调用这个函数
    $left_array = quick_sort($left_array);
    $right_array = quick_sort($right_array);
    //合并
    return array_merge($left_array, array($base_num), $right_array);
}
```

