# 算法

## 1.冒泡排序

 function bubble\_sort(array $arr){  
     foreach ($arr as $vv){  
         foreach ($arr as $k=>$v) {  
             if (($arr[$k] > $arr[$k + 1]) && isset($arr[$k + 1])) {  
                         $mid = $arr[$k + 1];  
                         $arr[$k + 1] = $arr[$k];  
                         $arr[$k] = $mid;  
            }  
        }  
    }  
     return $arr;  
 }

## 2.快速排序

 function quick\_sort($arr)  
 {  
     // 判断是否需要继续  
     if (count($arr) <= 1) {  
         return $arr;  
    }  
 ​  
     $middle = $arr[0]; // 中间值  
 ​  
     $left = array(); // 小于中间值  
     $right = array();// 大于中间值  
 ​  
     // 循环比较  
     for ($i=1; $i < count($arr); $i++) {  
         if ($middle < $arr[$i]) {  
             // 大于中间值  
             $right[] = $arr[$i];  
        } else {  
 ​  
             // 小于中间值  
             $left[] = $arr[$i];  
        }  
    }  
 ​  
     // 递归排序两边  
     $left = quick\_sort($left);  
     $right = quick\_sort($right);  
 ​  
     // 合并排序后的数据，别忘了合并中间值  
     return array\_merge($left, array($middle), $right);  
 }

## 3.排序算法的对比

 //冒泡排序（数组中实现）  
 function bubble\_sort($arr) {  
    $cnt = count($arr);  
    if($cnt<=0) return false;  
    for($i=0; $i<$cnt;$i++) {  
        for($j=$cnt-1; $j>$i;$j--) {  
            if($arr[$j]<$arr[$j-1]) {  
                $tmp = $arr[$j];  
                $arr[$j] = $arr[$j-1];  
                $arr[$j-1]=$tmp;  
            }  
        }  
    }  
    return $arr;  
 }  
 //快速排序（数组排序）  
 function quicksort($arr) {  
    if(count($arr)<=1) return $arr;  
    $key = $arr[0];  
    $left\_arr = array();  
    $right\_arr = array();//新建一个数组  
    for($i=1; $i<count($arr); $i++) {  
        if($arr[$i]<$key) {  
            $left\_arr[] = $arr[$i];  
        } else {  
            $right\_arr[] = $arr[$i];  
        }          
    }  
    $left\_arr = quicksort($left\_arr);  
    $right\_arr = quicksort($right\_arr);  
    return array\_merge($left\_arr,array($key),$right\_arr);  
 }  
 for($i=0;$i<10000;$i++) {//测试数组  
    $arr1[$i] = rand(0,10000);  
      
 }  
 $time\_start = explode(" ",microtime());  
 bubble\_sort($arr1);  
 $time\_end = explode(" ", microtime());  
 echo "冒泡排序法所需要的时间:";  
 echo ($time\_end[0]+$time\_end[1] - $time\_start[0]-$time\_start[1])."<br>";//测试冒泡排序法所需要的时间  
 echo '----------------------------------'.'<br>';  
 $time\_start = explode(" ",microtime());  
 quicksort($arr1);  
 $time\_end = explode(" ", microtime());  
 echo "快速排序法所需要的时间:";  
 echo ($time\_end[0]+$time\_end[1] - $time\_start[0]-$time\_start[1])."<br>";//测试快速排序法所需要的时间  
 echo '-------------------------------------'."<br>";  
 $time\_start = explode(" ",microtime());  
 sort($arr1);//php内置对数组排序函数  
 $time\_end = explode(" ", microtime());  
 echo "PHP内置函数排序法所需要的时间:";  
 echo ($time\_end[0]+$time\_end[1] - $time\_start[0]-$time\_start[1])."<br>";//测试PHP内置函数排序法所需要的时间  
 测试结果：  
 冒泡排序法所需要的时间:22.864297151566  
 ----------------------------------  
 快速排序法所需要的时间:0.22899389266968  
 -------------------------------------  
 PHP内置函数排序法所需要的时间:0.037806987762451  
 ​  
 ​  
 总结,最后使用php的内置函数sort,进行排序  
 ​

## 4.如何进行数组去重?

 使用array\_flip();作用是将数组的key和value进行交换.  
 $arr = array(2,2,3,3,3,1,321,432,1,2);  
 $arr=array\_flip($arr);  
 $arr=array\_flip($arr);  
 var\_dump($arr);//array(5) { [9]=> int(2) [4]=> int(3) [8]=> int(1) [6]=> int(321) [7]=> int(432) }  
 ​