

# 数组排序算法

## 1. 概述:

本节主要目的是让大家对数组排序算法有思维上的认识，或者说为了面试，而不是为了使用。为啥？因为不管那个平台的面向对象开发，系统都提供了相应的排序算法，而既然是系统所提供的，肯定是相对“最优”排序算法了，所以实际应用中并不需要我们亲自去实现一个具体的排序过程。

## 2. 系统排序函数

sort、asort、ksort、usort、arsort、krsort、shuffle

sort示例:

```
////////////////////////////////////
$arr5 = array("a" => "cc", "1" => "bb", 0 => "aa");
echo "排序前数组"; print_r($arr5). "<br>";
sort($arr5);
echo "排序后数组"; print_r($arr5). "<br>";
?>
```

值排序了，但，下标没了。。。

排序前数组Array  
(  
    [a] => cc  
    [1] => bb  
    [0] => aa  
)  
排序后数组Array  
(  
    [0] => aa  
    [1] => bb  
    [2] => cc  
)

asort示例:

```
////////////////////////////////////
$arr6 = array("a" => "cc", "1" => "bb", 0 => "aa");
echo "排序前数组"; print_r($arr6). "<br>";
asort($arr6);
echo "排序后数组"; print_r($arr6). "<br>";
echo "<hr>";
////////////////////////////////////
?>
```

哎呦，这次下标没错哦。。。

排序前数组Array  
(  
    [a] => cc  
    [1] => bb  
    [0] => aa  
)  
排序后数组Array  
(  
    [0] => aa  
    [1] => bb  
    [a] => cc  
)

更多排序，请查php开发手册

手册》函数参考》与变量有关的拓展》数组》sorting arrays

## 3. 冒泡排序

```
$arr = array(12, 4, 2, 9, 8, 3);
$n = count($arr);
for ($i = 1; $i <= $n - 1; $i++) { //比较趟数n-1, 每一趟结束冒一个泡
    for ($j = 0; $j < $n - $i; $j++) { //每趟需比较个数, 为上一趟的比较次数-1次
        if ($arr[$j] > $arr[$j + 1]) { //两两比较和交换
            $t = $arr[$j];
            $arr[$j] = $arr[$j + 1];
            $arr[$j + 1] = $t;
        }
    }
}

echo "结果为: "; print_r($arr);
```

冒泡耗时:2.21729278564E-5结果为:

```
//<<<<<<<选择排序>>>>>>>
$arr = array(12, 4, 2, 9, 8, 3);

$pre_time = microtime(true);
for ($i = 1; $i <= $n - 1; $i++) { //比较趟数n-1，每一趟结束将最大值放到后面
    $max = 0;
    $idx = 0;

    for ($l = 0; $l <= $n - $i; $l++) { //每趟需比较个数，为上一趟的比较次数-1次
        if ($arr[$l] >= $max) { //比较获得最大值
            $max = $arr[$l];
            $idx = $l;
        }
    }

    $t = $arr[$idx];
    $arr[$idx] = $arr[$n - $i];
    $arr[$n - $i] = $t;
}

$end_time = microtime(true);

echo "<pre>";
echo "选择耗时:".($end_time - $pre_time);
echo "结果为: \n"; print_r($arr);
echo "<br>";
```

```
Array
(
    [0] => 2
    [1] => 3
    [2] => 4
    [3] => 8
    [4] => 9
    [5] => 12
)
```

略