Unknown date | Unknown author

前端常见算法JS实现

算法是程序的灵魂，一个优秀的前端工程师对算法也是要有所了解的。

1. 排序算法

1. 冒泡排序

```
//冒泡排序
function bubbleSort(arr){
var i = j = 0;
for(i=1;i<arr.length;i++){
for(j=0;j<=arr.length-i;j++){
var temp = 0;
if(arr[j]>arr[j+1]){
temp = arr[j];
arr[j] = arr[j+1];
arr[j+1] = temp;
}
}
}
}
```

2. 快速排序

```
//快速排序
function quickSort(arr,l,r){
if(l < r){
var i = l, j = r, x = arr[i];
while(i<j){
while(i<j && arr[j]>x)
j--;

if(i<j)
//这里用i++，被换过来的必然比x小，赋值后直接让i自加，不用再比较，可以提高效率
arr[i++] = arr[j];

while(i<j && arr[i]<x)
i++;

if(i<j)
//这里用j--，被换过来的必然比x大，赋值后直接让j自减，不用再比较，可以提高效率
arr[j--] = arr[i];
}
arr[i] = x;
```

```
quickSort(arr, l, i-1);
quickSort(arr, i+1, r);
}
}
```

3. 二路归并

将两个按值有序序列合并成一个按值有序序列，则称之为二路归并排序

```
function merge(left, right) {
var result = [],
il = 0,
ir = 0;

while (il < left.length && ir < right.length) {
if (left[il] < right[ir]) {
result.push(left[il++]);
} else {
result.push(right[ir++]);
}
}
while(left[il]){
result.push(left[il++]);
}
while(right[ir]){
result.push(right[ir++]);
```