冒泡排序

ES5的写法

var arr=[10,20,2,9,85,201,32,2,45,4]

var len=arr.length;

for(let i=0;i<len-1;i++){

for(let k=0;k<len-i-1;k++){

if(arr[k]>arr[k+1]){

//重排

var a=arr[k+1];

arr[k+1]=arr[k];

arr[k]=a;

}

}

}

console.log(arr);

ES6的写法

var arr=[10,20,2,9,85,201,32,2,45,4]

var len=arr.length;

for(let i=0;i<len-1;i++){

for(let k=0;k<len-i-1;k++){

if(arr[k]>arr[k+1]){

//重排

[arr[k],arr[k+1]]=[arr[k+1],arr[k]];

}

}

}

这里用数组解构简化，而不用中间变量

[arr[k],arr[k+1]]=[arr[k+1],arr[k]];

# 链表

不同于数组，链表中的元素在内存中并不是连续放置的。每个元素由一个存储本身的节点和一个指向下一个元素的引用（指针）。

# 算法

pages/algorithm/algorithm.html

#### 二分查找O(n)

有序数组中查找值

## 归并排序（nlogn）

-----分治算法

将 大数组切分为小数组，直到每个小数组只有一个项

然后将 小数组归并成 有序的较大数组，一层一层归并。直到成为一个有序的大数组。

能大量减少比较次数，但是也考虑存在递归，函数调用的消耗。

# 链表

数组中插入，或删除项的成本很高，因为需要移动元素。