

拉勾教育

— 互联网人实战大学 —

《前端高手进阶》

朱德龙 前中兴软创主任工程师

— 拉勾教育出品 —

第27讲：到底该不该了解算法

算法是为了解决某个问题抽象而成的计算方法

可以简单地把算法比作一个拥有输入和输出的函数

这个函数总能在有限的时间经过有限的步骤给出特定的解



前端开发场景更关注页面效果及用户交互
涉及数据计算的场景比较少



前端运行环境浏览器性能有限



在多端的系统中算法运用在后端会更加高效



但随着 Web 技术的不断发展
前端运行环境以及 Node.js 的计算能力不断加强
算法将被用于更多的开发场景中





01

时间复杂度

运行时间



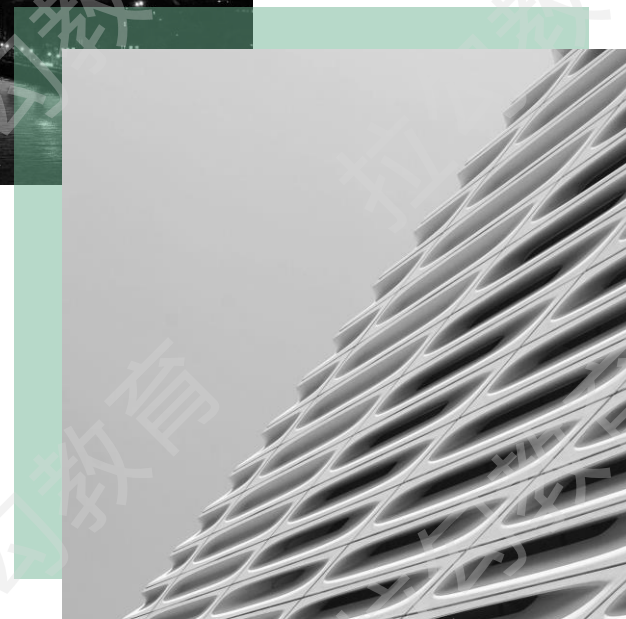
02

空间复杂度

存储空间

“复杂度”

可以理解为一个带有参数的函数
简写为 O
 O 的参数一般为 1 或 n 的表达式



```
arr.reduce((acc, cur) => acc += cur, 0)
```


$$S_n = na_1 + \frac{n(n-1)}{2}d, n \in N^*$$

等差数列求和公式

时间复杂度取决于 表达式执行次数



```
let tmp = arr[0]
for(let i=0;i<arr.length-1;i++){
  arr[i] = arr[i+1]
}
arr[arr.length-1] = tmp
```

```
let newArr = []  
for(let i=0;i<arr.length-1;i++) {  
  newArr[i] = arr[i+1]  
}  
newArr.push(arr[0])
```

简写原则



多项式组成的复杂度
取最高次项，并省略系数

比如 $O(3n+1)$ 简写成 $O(n)$



不同参数可以统一用 n 表示

比如遍历长度为 m 乘以 n 的数组
复杂度 $O(mn)$ 写成 $O(n^2)$

$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(x^n)$

TimSort 排序

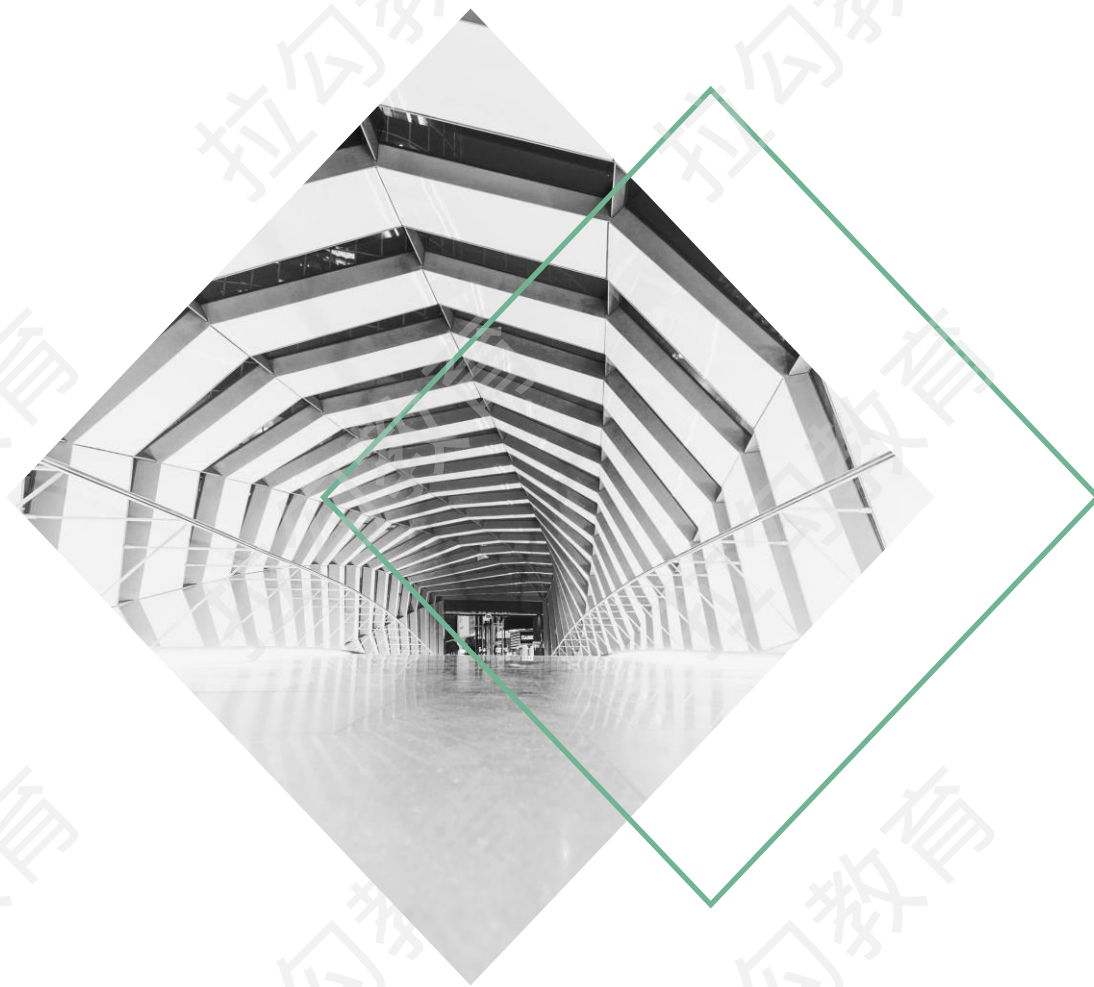
拉勾教育

— 互联网人实战大学 —

排序算法就是

让线性结构的数据按照升序或降序的方式排列的操作

是最基础也是使用频率较高的算法



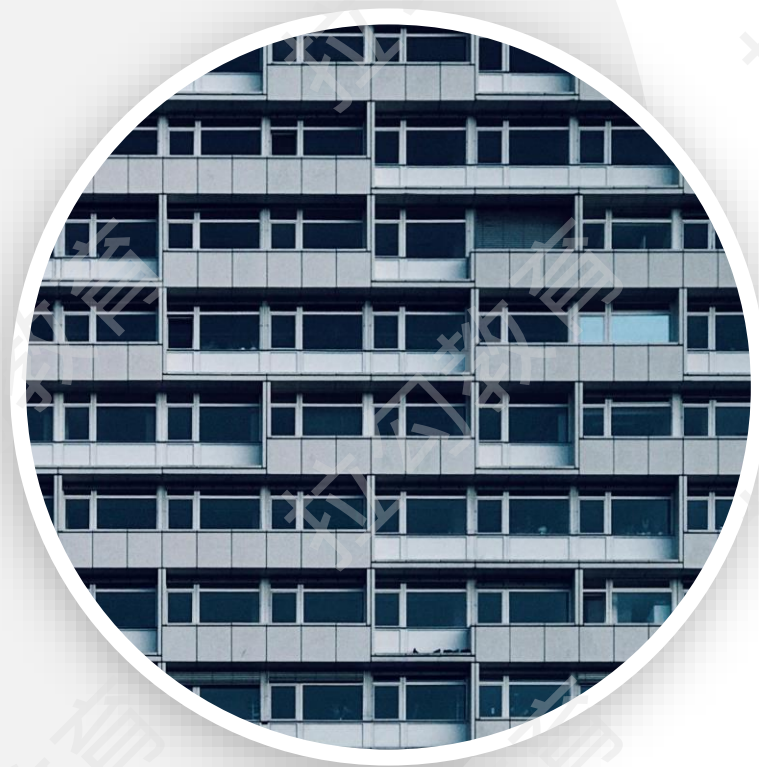
排序的意义在于可以大大减少后续操作的时间复杂度

排序这个基础的操作有多种算法



TimSort

在 Java、Python 等
多种语言环境广泛应用的排序算法



折半插入排序

归并排序

01

02

TimSort 排序

拉勾教育

— 互联网人实战大学 —

01.

根据数组长度进行计算
得到一个数值 `minRunLength`
表示最小的子数组 `run` 的长度

02.

通过 `while` 循环遍历数组
计算下一个 `run` 的长度

03.

判断 `currentRunLength`
和 `minRunLength` 的大小

06.

按次序合并 `pendingRuns`
中的 `run`，得到最终结果

05.

将 `pendingRuns` 中的部
分 `run` 进行合并

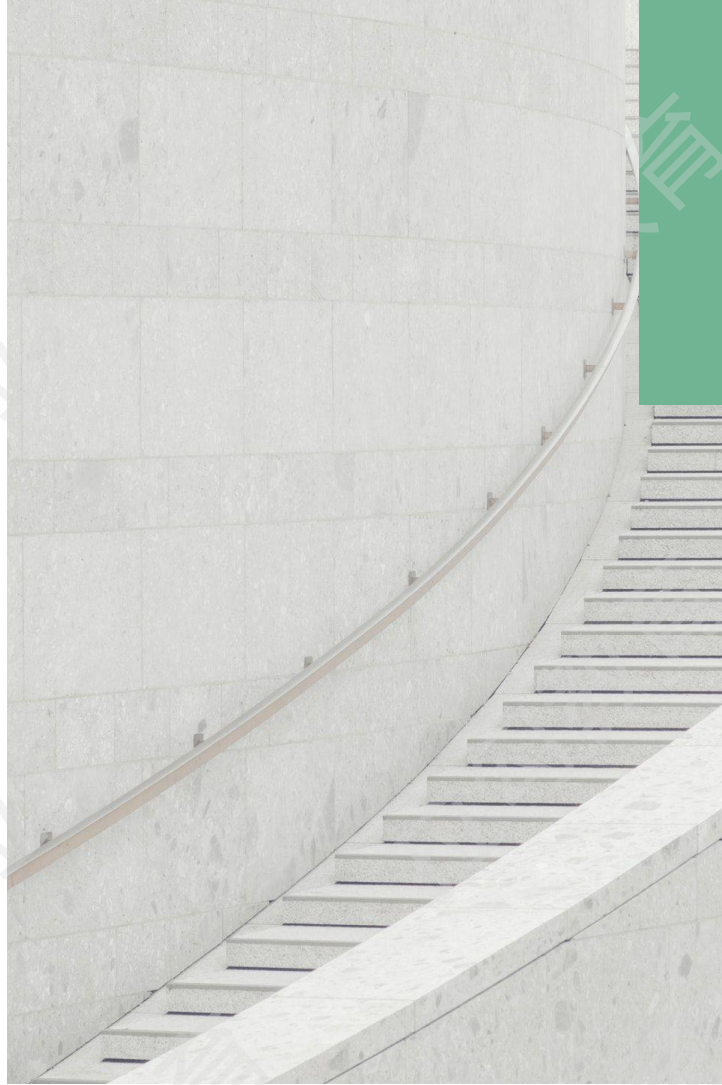
04.

将得到的 `run` 压入栈
`pendingRuns` 中
等待进一步的合并

补充 1：折半插入排序

折半插入

用折半查找插入点取代按顺序依次寻找插入点
从而加快寻找插入点的速度



拉勾教育

— 互联网人实战大学 —



补充 1：折半插入排序

拉勾教育

— 互联网人实战大学 —

```
function binayInsertionSort(arr) {  
  for (var i = 1; i < arr.length; i++) {  
    if (arr[i] >= arr[i - 1]) continue  
    let temp = arr[i];  
    let low = 0;  
    let high = i - 1;  
    while (low <= high) {  
      mid = Math.floor((low + high) / 2);  
      if (temp > arr[mid]) {  
        low = mid + 1;  
      } else {  

```

补充 1：折半插入排序

拉勾教育

— 互联网人实战大学 —

```
        low = mid + 1;
    } else {
        high = mid - 1;
    }
}
for (var j = i; j > low; --j) {
    arr[j] = arr[j - 1];
}
arr[j] = temp;
}
```

归并排序 (Merge Sort)

采用分治法 (Divide and Conquer) 的思想

把数组拆分成子数组

先对每个子数组进行排序

然后再将有序的子数组进行合并

得到完全有序的数组

补充 2：归并排序

拉勾教育

— 互联网人实战大学 —

```
function mergeSort(array) {  
  function merge(leftArr, rightArr) {  
    var result = [];  
    while (leftArr.length > 0 && rightArr.length > 0) {  
      if (leftArr[0] < rightArr[0]) {  
        result.push(leftArr.shift());  
      } else {  
        result.push(rightArr.shift());  
      }  
    }  
    return result.concat(leftArr).concat(rightArr);  
  }  
}
```

补充 2：归并排序

拉勾教育

— 互联网人实战大学 —

```
else
    result.push(rightArr.shift());
}
return result.concat(leftArr).concat(rightArr);
}
if (array.length == 1) return array;
var middle = Math.floor(array.length / 2);
var left = array.slice(0, middle);
var right = array.slice(middle);
return merge(mergeSort(left), mergeSort(right));
}
```

介绍了算法的两个重要效率指标：**时间复杂度**和**空间复杂度**

重点分析了 JavaScript 的 `Array.prototype.sort()` 函数的

底层实现算法 **TimSort**



你在开发过程中还用到过哪些算法



Next: 第28讲: 《你都了解过哪些编程方式? 》

拉勾教育

— 互联网人实战大学 —



下载「拉勾教育App」
获取更多内容