

2年经验社招微信5面面经

最近才有空整理一下面经，加上HR面一共6面，前前后后大概一个半月的时间

一面（组员面）

算法

1. 洗牌算法：对52张牌洗牌，要求尽量洗乱，而且原牌不能在原位置上重复

```
1 function shuffleArray(array: any[]){
2
3 }
```

供参考的解法：[blog.csdn.net/qq_26399665...](https://blog.csdn.net/qq_26399665)

在leetcode上面也有类似的题目：[leetcode-cn.com/problems/sh...](https://leetcode-cn.com/problems/shuozhi-zhe-xing-shi-bu-ke-xing/)

1. 给定一个字符串，编写一段代码测试该段字符串的括号是否完全闭合。

```
1
2 // [()]{}{[()()]()} - true
3
4 // [(()] - false
```

这道题也算很经典了，leetcode传送门👉 [leetcode-cn.com/problems/va...](https://leetcode-cn.com/problems/valid-parentheses/)

1. 现有新旧两个数组，旧数组经过一系列操作(包括新增、删除、移动等)后可得到新数组。请实现一个函数，找出旧数组上应该执行的操作列表。

```
1
2
3 const nodes = ['A', 'B', 'C', 'D'];
4
5 const after = ['B', 'A', 'E', 'D', 'C'];
6
7 const operations = [];
8
9
10
```

```

11 // 请自行定义VDomElement结构
12
13 function diff(nodes, after) {
14
15
16
17
18 }
19
20
21

```

思想类似于dom diff

1. 数组a[N]，存放了数字1至N-1，其中某个数字重复一次。写一个函数，找出被重复的数字。时间复杂度必须为O(N), 空间复杂度不能是O[N]。函数原型：int find(int a[], int N)
 - a. 有一个二叉树，每个节点的值是一个整数。写一个函数，判断这棵树中是否存在从根到叶子节点的一个路径，这个路径上所有节点之和为某一个值。存在返回1， 否则返回0。

```

1 struct TreeNode
2 {
3     int value;
4     struct TreeNode *left, *right;
5 };
6 int haspath(struct TreeNode *root, int value)
7

```

leetcode传送门👉 leetcode-cn.com/problems/pa...

1. 最长上升子序列 leetcode-cn.com/problems/lo...

面试

1. 问项目相关的问题
2. vue父子组件如何通信
3. 用webpack做过哪些优化（开发时候，打生产包）
4. tree-shaking原理 juejin.cn/post/684490...
5. node和浏览器 事件循环的不同之处 www.yuque.com/tency/cq3tf...
6. 讲下vue 为什么要用虚拟dom ,优缺点是？ 性能。
7. dom diff如何比较？ zhuanlan.zhihu.com/p/362539108 。 www.cnblogs.com/wind-lanyan...
8. 讲下防抖跟节流的原理和应用场景。

9. 讲下你对闭包的理解，以及闭包的应用场景。

10. 讲下http和https的不同之处。

11. 有了解过webassembly吗？

二面（组长面）

笔试：

LRU算法 leetcode传送门👉 [leetcode-cn.com/problems/lr...](https://leetcode-cn.com/problems/lru-cache/)

面试：

1. 项目相关

2. 浏览器事件循环

3. hybrid开发js与原生交互的原理

虽然问的问题不多，但是每个问题问的都挺深

三面（总监面）

算法：

三数之和

给定一个包含 n 个整数的数组 $nums$ ，判断 $nums$ 中是否存在三个元素 a ， b ， c ，使得 $a + b + c = 0$ ？
找出所有满足条件且不重复的三元组。

注意：答案中不可以包含重复的三元组。

例如，给定数组 $nums = [-1, 0, 1, 2, -1, -4]$ ，

满足要求的三元组集合为：

```
1
2  [ [-1, 0, 1],
3
4  [-1, -1, 2]
5
6  ]
```

概率题：

某城市有两种颜色的出租车：蓝色和绿色（市场占有比例为15：85）。一辆出租车夜间肇事后逃逸，但还好当时有一位目击证人，这位目击者认定肇事的出租车是蓝色的。但是，他“目击的可信度”如何呢？公安人员在相同环境下对该目击者进行“蓝绿”测试得到：80%的情况下识别正确，20%的情况不正确。请问可以算出在有目击证人情况下肇事车辆是蓝色的概率为多少？

面试：

1. 项目相关问题,一直深挖

四面（交叉面）

算法：

1. 设定关系（A=1,B=2,C=3, ..., Z=26,AA=27,AB=28,..., AAA=XXX, ...），写一个转换函数，根据上面规则把一个字符串转换为数字。

```
const convert = (str) =>{
  let res = 0;
  const strArr = str.split('');
  let j;

  const { length } = strArr;
  for(let i = length - 1, j = 1; i >= 0; i--, j *= 26){
    const value = strArr[i].toUpperCase();
    if( value < 'A' || value > 'Z'){
      return 0;
    }
    //value.charCodeAt(0) - 64可以算出A=1,B=2,C=3等对应关系
    res = res + (value.charCodeAt(0) - 64) * j
  }
  return res
}
```

1. 二叉搜索树的第k大节点 [leetcode-cn.com/problems/er...](https://leetcode-cn.com/problems/er-cha-sou-suo-shu-de-di-k-da-jie-dian/)

题目的附加条件是不能二叉树全量存储到另外的存储空间。

所以不能直接中序遍历再去取数组的倒数第三个元素。

```

const kthLargest = (root,k) => {

    let kthValue = 0;

    const helper = (root) => {
        if(!root){
            return;
        }
        helper(root.right);
        if(!--k){
            kthValue = root.value;
            return;
        }
        helper(root.left);
    }
}

```

1. 从一个整数数组中，找出位置连续的最长递增序列
2. 单链表中间位置的节点

```

//双指针，快指针每次走两步，慢指针每次走一步
const middleNode = (head) =>{
    let slow = head;
    let fast = head;

    while(fast && slow){
        slow = slow.next;
        fast = fast.next.next;
    }
    return slow;
}

```

1. 两个有序（从小到大）单链表{value:Number;next:Object}，找出这两条链表中value相等的节点，构造成一条新的链表。

```

const commonNodeList = (l1,l2) => {
  const newNodeList = new ListNode(0)
  //哪个l的value比较小就往next移
  while(l1 && l2){
    if(l1.value === l2.value){
      newNodeList.next = l1.value;
      l1 = l1.next;
      l2 = l2.next;
    }else if(l1.value < l2.value){
      l1 = l1.next;
    }else{
      l2 = l2.next;
    }
  }
  return newNodeList
}

```

1. 反转数组，在原数组反转（不使用arr.reverse方法）

```

const ownReverse = (arr) =>{
  const { length } = arr;
  for(let i = 0, j = length - 1; i < j; i++,j--){
    [arr[i],arr[j]] = [arr[j],arr[i]]
  }
  return arr
}

```

面试:

项目相关

延伸出H5如何调用原生方法

在项目中的性能优化相关的手段

五面（GM面）

面试:

项目相关（成长最快的一段时间以及原因，项目迭代周期，重构原因，测试周主要工作，自动化测试）

也问了H5如何调用原生方法

开放性问题：

假设现在有一个web版的微信，使用扫码进行登录的原理，延伸到除了轮询还有什么其他方式，http1.1通过哪个字段可以设置长链接

假设微信好友上限是10000，那如何保证好友列表渲染的性能

对于好友头像（大量小图片的加载）有什么优化手段

在好友列表中，还需要有个好友搜索的功能，模糊搜索，如何在不用所有好友进行一次遍历的情况下找到匹配的人名？（哈希表？）

大文件如何实现断点续传？

如何实现秒传？（相同的文件不需要重新再发一次），延伸至如何判断文件内容是否相同？

每轮面试都会遇到与算法相关的题目，所以算法非常之重要，还有项目中遇到的最大挑战，还有从项目中延伸的跟基础相关的一些问题，项目的話也很注重深度，所以项目和算法都是需要去好好准备的，其他的前端基础相关的也挺多考察的，也蛮注重知识深度的，会一直深挖背后的一些问题。

不过最后GM面没有过,仍需努力😞😞😞😞😞😞