

广州大厂前端面试题

字节

一面：

二面：

Fordeal

一面

映客直播

玄武科技-D轮公司

易工品

有米科技

bigo

一面：

二面：

三面：

cvte

一面：

荔枝

二面 扩展性问题很多 50min

三面技术面 扩展性问题依旧很多 50min

四面 部门负责人 扩主要是聊一些项目上的问题 30min

cvte

豌豆思维

lineshop

映客

一面：

二面：

荔枝

一面：

二面：（全是问简历的内容）

hr 面：

虾皮

一面 90min 感觉面试官很nice 少数聊的不错的面试官 问的相对比较深入

字节飞书

企业微信

网易游戏

阿里文娱

汇量科技

网易云音乐

一面：

二面：

腾讯一面

凡科笔试

字节

1. 什么是协商缓存与强缓存；
2. 讲讲客户端存储的方式有哪些；
3. 如何防止脚本获取cookie；
4. 讲讲状态码301/302/307的区别；
5. 讲讲csrf、xss网络攻击；
6. 如何解决前端跨域如何解决前端跨域；
7. js 事件循环模型；
8. 实现模板字符串；
9. 讲讲vue双向绑定原理；
10. 讲讲vue的计算属性为什么说是无害的；

一面：

1. 假设按照升序排序的数组在预先未知的某个点上进行了旋转。
(例如，数组[0,1,2,4,5,6,7]可能变为[4,5,6,7,0,1,2])。搜索一个给定的目标值，如果数组中存在这个

目标值，则返回它的索引，否则返回-1。你可以假设数组中不存在重复的元素。你的算法时间复杂度必须是 $O(\log n)$ 级别。

2. 【笔试】求以下输出

JavaScript | 复制代码

```
1 ▾ async function async1(){
2     console.log('async1start');
3     await async2();
4     console.log('async1end');
5 }
6 ▾ async function async2(){
7     console.log('async2');
8 }
9 console.log('scriptstart');
10 ▾ setTimeout(function () {
11     console.log('setTimeout');
12 }, 0)
13 async1();
14 ▾ new Promise(function (resolve) {
15     console.log('promise1');
16     resolve();
17 })
18 ▾ .then(function () {
19     console.log('promise1then');
20     return 'promise1end';
21 })
22 .then((res) => { console.log(res); })
23 .then((res) => { console.log(res); });
24 console.log('scriptend');
```

3. BFC
4. 浏览器渲染流程
5. XSSCSRF
6. 手写节流
7. 性能优化webpack打包过程
8. HTTP2特性浏览器缓存
9. 最近学什么前端技术
10. 不定宽高的元素如何水平垂直居中?
11. 有没有了解过 vue 源码?
12. 知道 vue 的生命周期吗?
13. 知道 vue 的双向绑定如何实现的吗?
14. 用过 vuex 吗? 哪些情况用 vuex? 除了 vuex , 还有哪些组件间通信方式? 你是如何处理兄弟组件

间通信的？

15. 有使用过 ES6 的 decorator 吗？
16. 有了解过 koa 源码吗？
17. 有自己写过 koa 中间件吗？
18. 有了解过 webpack 源码吗？
19. 浏览器跨域的方式有哪些？
20. <http://ip:port> 与指向该 ip:port 的域名是同域吗？
21. 有做过哪些首屏性能优化吗？
22. 标准盒模型与怪异盒模型
23. 数据库索引的定义？作用？如果一个表修改频繁，那还要建立索引吗？为什么？
24. 你们连接数据库使用的什么？有用过 RPC 吗？
25. 什么是事物？事物的分类？
26. 有这些表：学生表、班级表、课程表、学生选课表（省略字段）。请写出如下 sql：查询平均成绩在 60 分以上的学生的 id、平均成绩。
27. [笔试]有一个单向链表，奇数位升序，偶数位降序，请将它转换为一个整体升序的单向链表。

二面：

1. 自我介绍；
2. 项目；
3. Q: B/S C/S 架构的定义是什么？
4. 为什么 ref 等原生模块需要针对 electron 重新编译？
5. electron 的架构是什么？
6. 如果有多个渲染进程，他们之间可以怎样进行通信？
7. exec 和 spawn 的区别是什么？
8. Node 架构是怎样的？
9. 硬链接和软链接的区别是什么？
10. 你为项目做了哪些优化？
11. webpack import 打包后的结果是怎样的？
12. 懒加载加载的的时机是什么时候，怎么加载的？
13. treeshaking 是怎样实现的？
14. 为什么 treeshaking 只能用于 ES6 模块？发生在哪个阶段？
15. 多进程打包是怎么做的？为什么选择使用 ES？
16. pdf 解析是怎样做的？
17. 描述一下项目的数据库设计 未来的职业规划是什么？为什么想做全栈？

18. 现有一个含有字符串的数组，形如：["ab","c","ab","d","c"]要求将其中出现的重复字符串，依次添加数字序号，如：["ab1","c1","ab2","d","c2"]
- 要求：
- 1).未重复出现的字符串不处理；
 - 2).仅对相同的一组字符串依次添加序号，而不是共用一组序号；
 - 3).保持原数组顺序；
19. 你的优缺点；
20. 你有什么想问我的：没有。
21. 【笔试】求以下输出

JavaScript | 复制代码

```
1  function Person() { }
2      let p = new Person();
3      console.log(
4          p.__proto__
5          .__proto__
6          .constructor
7          .prototype
8          .constructor
9          .constructor
10         .constructor
11     );
```

【笔试】有如下布局，问 first-child last-child 宽度分别是多少

```
1  .container {  
2      display: flex;  
3      width: 600px;  
4      height: 300px;  
5  }  
6  
7  .first-child {  
8      width: 500px;  
9      flex-shrink: 1;  
10 }  
11  
12 .last-child {  
13     width: 400px;  
14     flex-shrink: 2;  
15 }
```

【笔试】回答以下代码运行结果，并详细描述 node 事件循环过程

```
1  // --- 1 ---  
2  setTimeout(() => {  
3      console.log('timeout');  
4  }, 0);  
5  setImmediate(() => {  
6      console.log('immediate');  
7  });  
8  // --- 2 ---  
9  fs.readFile('xxx', (err, res) => {  
10     setTimeout(() => {  
11         console.log('timeout');  
12     }, 0);  
13     setImmediate(() => {  
14         console.log('immediate');  
15     });  
16 });
```

【笔试】二选一

1. 自己实现一个 Promise.all
2. 已知两个单向链表某个节点相同，现在给出两个链表的头节点，请找出两个链表相同的节点

Fordeal

一面

1. 项目相关
2. webview内嵌网页，在发布之后，还会出现旧的页面，也就是资源未更新。
3. 滚动穿透如何实现
4. URL请求过程
5. 项目上遇到困难一个
6. 现场手撕代码 `add(1)(2)(3)(4) => 输出10`
7. 八股文，忘记了
8. 排平数组
9. 实现一个函数，参数是wait 等待时间 func 需要执行的函数 times 次数。等待wait时间然后再执行func，这样执行times次。

映客直播

1. 常用的ES6特性
 - let / const、箭头函数、解构赋值、默认参数、扩展运算符、类、字符串模板、数组新增的高阶函数、Promise
2. JavaScript的数据类型
 - 感觉念不完全
 - Object Null Number Undefined String Boolean Symbol BigInt
3. 箭头函数和普通函数的区别——this指向todo
 - 总结：
 - this 引用的是定义箭头函数的上下文；
 - 箭头函数不能使用arguments、super 和new.target，也不能用作构造函数。此外，箭头函数也没有prototype 属性。
 - 箭头函数不能使用arguments。如果在箭头函数内部使用了arguments则会报错如下。

- 在箭头函数中，this 引用的是定义箭头函数的上下文。也就是外层啦
- 不能使用new.target，也不能做构造函数，也没有prototype属性

4. HTTP缓存

5. 强缓存和弱缓存

6. 浏览器本地存储及其区别

7. SSR原理

8. Vue的生命周期

9. 组件渲染的错误监听

10. 项目的优化——这一点需要更新简历，并且再组织下语言（done）

佰锐科技

11. SPA页面和传统页面的区别

12. 路由的两种模式：history和hash，hash的原理是什么

13. vue的生命周期

14. slot了解吗？有哪几种类型

15. 原型链 以及如何判断属性是否为对象自己的？

16. vuex的action和mutation

奥凯信息服务公司

17. 为什么不用vuex呢

18. 除了业务方面，前端可以做哪些优化？

19. 你做了前端哪些优化？

20. vue的diff算法

21. vue的数据双向绑定

22. Set、Map、Symbol数据结构需要学习

23. 原型链

玄武科技—D轮公司

上去就做了一张两开的试卷，我去。前面是试卷题目，后面是问的

1. 重排重绘的区别以及优化方法

2. 301和302的区别以及使用场景

3. 箭头函数和普通函数的区别
4. BFC以及触发BFC
5. get和post的区别
6. vue组件的通讯方式
7. 手撕new操作（new 构造函数）
8. 手撕promise.all
9. 手撕防抖和节流
10. webpack打包过程特别慢咋办
11. 给出一个字符串，找到重复次数最多的那个，并返回次数
12. 手撕快排，分析快排的过程以及使用场景
口说的
13. vue组件通讯方式只有这几种吗？
14. vue的v-for的key是怎么回事？diff的过程
15. 用过angular吗？
16. vue的虚拟列表
17. vue的router权限控制
18. 项目 阅读做的加载优化
19. 项目 懒加载的原理
20. 有没有做过组件库、脚手架工具之类的？

易工品

这家公司面试都比较简单，记录一下没有回答好的题目

1. vue3和vue2的区别
2. 你了解我们公司吗？——面试之前一定要了解好人家的公司！不然到嘴的鸭子飞了

有米科技

不用做笔试题，问的问题也很婉转巧妙。（不分先后顺序 记不得了）

一面一个小时

1. async和defer的区别

2. performance的FCP是什么
3. JavaScript的数据类型
4. commonjs的模块机制和es6模块机制
5. 浏览器支持的模块机制有哪些
6. UMD是啥
7. csrf攻击是什么以及如何预防
8. 深拷贝
9. 中间人攻击是什么？
10. 浏览器的缓存机制
11. md5是什么？
12. 做过容灾吗？
13. 项目部署的具体流程？
14. 有自己写过gitlab的yml文件吗？
15. docker部署SPA项目的劣势？
16. vue明明有数据劫持还要vdom 和 diff算法？
17. 常见的状态码
18. 有80多个HTTP请求，在非暴力的情况下，如何做到有序发送？（发布订阅模式）
19. 垂直居中布局如何实现
20. 了解过BFC
21. 开放性问题，有没有做过前端工程化，不限于脚手架、webpack、组件库等
二面（顺序不一定，想不起来了）20min
22. 在团队承担的角色，有没有带过新人
23. BFF了解吗？为什么用Node.js作为BFF的技术栈
24. 为什么还要用Nuxt.js框架？
25. 跨端交流的例子
三面（顺序不一定，有HR的问题）30min
26. 职业规划
27. 别人如何评价你的性格
28. 缺点优点
29. 比同事好在哪里
30. 期望薪资
31. 还有很多，忘记了

bigo

一面：

- 1.https加密协议
- 2.node worker, 僵尸进程
- 3.ts中, error和unknown的区别
- 4.算法：
 - a.求五位数字中所有的回文数
 - b.实现函数add(1)(2)(3)参数之和
 - c.给定一个整数n, 计算所有小于等于n的非负整数中数字1出现的个数。
例：n=13, 数字1出现在1, 10, 11, 12, 13中, 共6次, 返回6
注意：不要使用转字符串的方式, 时间/空间复杂度尽量低。
 - d.版本号如1.1.0, 1.2.1, 现有两个版本号a, b, 判断a相对于b是否是一个新版本
- 5.重写console.log方法, 实现每次打印结果之前增加一个打印时间戳
- 6.两列布局, 左边固定宽度, 右边是硬剩余的宽度

二面：

- 1.babel如何编译
- 2.如何避免团队成员写ts使用any的坏习惯
- 3.commonJs和es6模块加载有什么区别, 如果要同时支持使用, 如何实现
- 4.单元测试如何测试
- 5.typescript的最佳实践
- 6.前端如何分层设计的
- 7.重新设计现有的这个项目, 你会如何设计
- 8.各种团队实际操作问题
- 9.项目如何打包部署, 有什么不好的地方, 有没有推进去解决
- 10.为什么想跳槽
- 11.还有在面试其它公司吗
- 12.babel代码转换如何支持兼容性问题
- 13.如何实现打包分类
- 14.手写算法

三面：

- 1.tcp发送和接收包的情况，是一次性发，还是逐次发，如何排序，具体标志位是什么。
- 2.http 1.0、1.1、2.0分别解决什么问题，有什么特点
- 3.php项目移到node.js如何进行单元测试的保证，通过接口请求结果的幂等性验证
- 4.obj读取数据快的原因，js是怎么实现的
- 5.手写算法
- 6.聊人生

cvte

一面：

a.自我介绍

- 1.目前情况(工作毕业，期望工作情况，offer情况，是否考虑，看什么书，加班，工作强度)
- 2.能够接受什么技术栈
- 3.个人职业规划
- 4.期待薪资

b.数据结构

- 1.平时有用哪些数据结构，用在哪里

c.网络：

- 1.浏览器打开 baidu.com，到页面完全显示，中间发生什么
(网络请求过程，页面渲染过程)
- 2.http2 有了解过吗，http1 1.1发展特点区别
- 3.浏览器缓存，强缓存和协商缓存
- 4.三次握手，4次挥手

5.XSS, CSRF

c.js

- 1.this的指向
- 2.es6新特性，箭头函数(什么情况不使用)，proxy，装饰器，可选链，如何遍历一个对象
- 3.MVVM，如何进行数据劫持
- 4.熟悉移动端适配方案
- 5.闭包
- 6.数据类型，如何判断
- 7.setTimeout不能准时回调(浏览器和node区别)

8.如何实现new一个对象

d.css

1.BFC

f.基础

工程化

性能优化

设计模式和设计法则

混合APP通信(APP是H5混合)

原理

node(koa原理)

业务和团队推动能力

项目最难的地方

组件库有做抽取复用处理。

荔枝

- async和defer的区别
- 用css写出3种隐藏元素的方法
- 上面的方面有什么区别
- 手撕防抖代码
- 八股文输入输出题目——主要分清setTimeout和promise的执行时机

二面 扩展性问题很多 50min

- 从url到渲染页面题，尽可能详细。然后面试官会从每个阶段进行扩展性提问
- 浏览器缓存机制
- 浏览器白屏
- 重排重绘及其预防措施
- 项目的两点
- HTTPS如何保证安全性
- TLS握手过程
- eventloop
- jsBridge底层通讯的原理

- vue和react的区别

还有很多扩展性、讨论性的项目问题

三面技术面 扩展性问题依旧很多 50min

二面结束之后，面试官说他去叫HR，我以为直接进入了HR面试，直到人家问我模块机制的时候才发现这是个技术面！！！！

- 项目亮点、难点
- 模块机制 CMD了解吗？
- CMD和AMD的区别
- 除了webpack之外的其他打包、构建工具还用过哪些
- 富文本编辑遇到了什么坑？
- webpack的组件按需引入是webpack自己的吗？还是es6语法
- 抛弃框架和ES6语法，现场手撕一个确认弹窗？写到一半让讲思路
- webpack的分包如何实现？可以说下具体的配置吗？
- 常见的时间复杂度排序
- 如何做技术选型

不记得了.....

四面 部门负责人 扩主要是聊一些项目上的问题 30min

- CICD的流程？
- 机器如何部署的？
- 了解过集群吗？
- 如何区分国内、国外用户？
- 项目场景的假设
- 不记得了....

五面 HR面 30min+

主要是了解公司情况、部门情况以及HR的压力面试和试探、以及薪酬

- 项目亮点、难点
- 抗压能力的体现

给大家分享一下昨天面试经历

cvte

- 一开始就问项目，最难的地方，问的特别细
- https 原理，TLS握手过程
- vue 和 react diff
- react setstate 之后的流程
- v8 GC 原理
- 闭包
- 你是如何做项目优化的
- mvvm 理解
- 有学习什么新技术
- 请求20个接口都对第一屏内容很关键如何做优化
- 职业规划

豌豆思维

- 整个过程也是围绕项目问
- 如何优化项目构建，难点，sso 登录原理
- 讲讲 cicd 自动化部署
- vue 数据劫持
- vue 是如何进行收集依赖然后更新的
- vue key 的作用，纯数组如何设置key
- react setstate 什么时候是异步什么时候是同步
- 作为前端优势是什么

面试总结：

- 1、简历写的每个点都要经得起深挖
- 2、面试前一定要整理面试知识点和组织好语言，逻辑清晰，（关键词好好记）反复练习
- 3、可以内推千万不要走平台

lineshop

1. 微前端
2. 做什么性能优化（图片做了什么优化）
3. vue 数据劫持
4. ts
5. react setState 同步异步
6. fiber 什么时候不能暂停任务
7. tree-shaking 副作用
8. SSR 原理
9. 闭包原理和应用（react哪里应用到）
10. 可以隐藏 div 的所有方式 以及性能优化层面上考虑
11. ES6 特别看好的特性（装饰器和proxy）
12. proxy 是否可以拦截函数被调用
13. vue 3 跟 react hook 相似的 api 对比
14. 微任务有哪些
15. node eventloop 与浏览器的区别
16. 首屏渲染优化

映客

一面：

1. 讲一下 webpack5 的模块联邦的原理（公共依赖升级咋办）
2. webpack loader 和 plugin 的区别
3. webpack 文件指纹（hash，重新打包就会变，chunkhash，入口文件变就会变，contenthash，代码变了就变）
4. webpack 构建优化
5. promise 讲一下理解和状态
6. 宏任务和微任务讲一下
7. 做了哪些构建优化和性能优化
8. 讲一下 http 缓存，什么时候用协商缓存（no-store 和 no-cache 的区别）
9. 讲一下 vue 数据劫持

二面：

1. 讲一下你项目，有哪些数据可以说明你说的价值
2. 你在团队是一个什么样的角色
3. https 原理
4. http 连接整个过程
5. 一次http连接一定会4次挥手吗
6. 讲一下 http3, UDP 原理
7. get 和 post 本质的区别
8. 404 是客户端的错误还是服务端错误，502、302、304、306 分别指什么
9. 一千万条整型类型数据存储中如何找到出现次数最多的值
10. 其他忘了

hr面：

就问一下期望，谈薪相关

荔枝

一面：

1. h5 怎么适配移动端和 pc 端，rem 具体实现
 2. 一个梯型如何实现
 3. 你是如何通过构建优化达到启动速度快 5 倍，如何配置分包
 4. vue3 新属性
- 为什么要引入 composition api
 - proxy 比 Object.defineProperty 对比，proxy 可以劫持到第二第三层级吗
 - vue2 如何监听数组的变化
1. 有用过缓存吗
- http 浏览器有哪些缓存
 - js, css, 图片适用于什么缓存

1. 一个 tcp 可以发起多少个 http 请求? http2.0 有限制吗
2. 什么是 http 头部堵塞
3. http2 有什么缺点? 什么情况下会出现 tcp 堵塞, 网络不好为什么会造成堵塞?
4. 如何检测网站的性能? 除了谷歌工具如何通过代码检测, 如何分析某个包有哪些依赖
5. 你如何做网站的性能优化
6. 如何做到某个包点击才加载, 是如何实现按需加载的
7. 我们 react 组件一定要引入 react

二面：（全是问简历的内容）

1. 讲一下webpack5 模块联邦, 当时为什么要用它
2. 还了解 webpack 5 什么新属性
3. umd 和模块联邦对比有什么区别
4. webapck 的垫片和 preset 有什么区别, 如何做到不是全局方式引入垫片 (helper)
5. vue nextTick 原理
6. react key 和 vue key 的作用
7. 平时用什么布局, flex 的组成和原理, 除了vw、vh还有什么可以设置像素百分比
8. scss 的 scope 如何做到样式隔离
9. 讲一下你说的自选模板cli是怎么实现的
10. 你 git 管理是用什么工作流
11. 算法时间复杂度大小排序 n 、 $\log n$ 、 n^2 、 $2n$ 、 2^n
12. 如何让团队规范协同
13. 你想要一个什么样的环境
14. 还没问道你的点

hr 面：

1. 最近两家只待了一年就离职的原因
2. 你觉得在哪家公司收获最大
3. 如何在哪些事件可以表现你的抗压能力
4. 对加班的看法
5. 对荔枝的了解
6. 有没有拿到其他offer

虾皮

一面 90min 感觉面试官很nice 少数聊的不错的面试官 问的相对比较深入

- Node server内存机制
- V8的垃圾回收机制
- 内存泄漏有哪些场景？如何查找？
- 项目得nuxt.js服务流量处理？100万用户如何处理？
- 项目nuxt.js的缓存是如何做的？
- 网页的多国语言处理
- 了解前端缓存吗？
- 浏览器缓存？以及优先级？
- HTTP2的特性？服务器推送了解吗？
- HTTP3了解吗？和HTTP2、HTTP1的区别
- TLS的握手过程？哪个环节用到了私钥
- 打包工具？webpack的原理？
- 热更新具体描述一些？
- webpack压缩如何做的？
- babel如何转es6的代码？
- 跨域方案？
- 项目的错误监控方案？如果sentry的js文件加载不出来如何错误上报？

笔试

```
1 // 1、八股文
2 ▼ setTimeout(() => {
3     console.log('start');
4 ▼     Promise.resolve().then(() => {
5         console.log('Promise 1');
6 ▼         setTimeout(() => {
7             console.log('setTimeout 2');
8         })
9     });
10 ▼     setTimeout(() => {
11         console.log('setTimeout 1');
12 ▼         Promise.resolve().then(() => {
13             console.log('Promise 2');
14         });
15     });
16 }, 0)
17 console.log('end');
18 // 2、超级八股文
19 var s = 0;
20 var i = 1;
21 var funcs = [];
22 var n = 3;
23 ▼ function x(n) {
24 ▼     for (i = 0; i < 3; i++) {
25 ▼         funcs[i] = () => {
26             s = s + i * n;
27             console.log(s);
28         };
29     }
30 }
31 x(1);
32 funcs0;
33 funcs1;
34 funcs2;
35 // 3、编程题 实现一个循环检测的函数?
36 ▼ const a = {
37 ▼     b: {
38         c: {}
39     }
40 };
41 a.b.c.d = a;
42 console.log(isCycle(a));
43 // 4、斐波那契数列数列
44 /*
45     问题描述:
```

```

46      假设你正在爬楼梯。需要 n 阶你才能到达楼顶。
47      每次你可以爬 1 或 2 个台阶。你有多少种不同的方法可以爬到楼顶呢？
48      注意：给定 n 是一个正整数。
49      实例：
50      输入： 3
51      输出： 3
52      解释： 有三种方法可以爬到楼顶。
53      1. 1 阶 + 1 阶 + 1 阶
54      1. 1 阶 + 2 阶
55      1. 2 阶 + 1 阶
56      */

```

字节飞书

一上来就手撕题，有一题撕不出来有点尴尬

- leetcode 31
- 手撕发布订阅模式 EventEmitter

问的大部分是项目延展问题

- 拿到一个需求功能，你的开发步骤
- SEO怎么做的
- 如果有一个用户白屏咋整？（讨论题）
- 印象深刻的故障解决流程
- 技术选型是怎么做的？解决了什么问题？

企业微信

太惨烈了都忘记问了什么，只记得几个问题 面了一个半小时

- 用过PWA吗？
- SSR的原理
- 同源两个标签页的通讯方式
- 做题

1. 二维平面上有 N 个点，请找出两两之间斜率最大的两个点

```

1  ▼ interface Point {
2      x: number;
3      y: number;
4  }
5  ▼ function run(arr: Point[]): [Point, Point] {
6      // TODO
7  }

```

1. 有一个 4x4 的矩阵，上面随机分布了 9 个点。你每次可以把一个点移动到其他格子，请移动最少的步数，另矩阵内有横、竖、斜三条线能连上 4 个点：

0000

0000

0XXX

XXXX

如上述例子中，"o" 为点，"x" 为空位，则最少移动两个点：

0000

0X0X

00XX

0XXX

在第一行、第一列，正斜线都能连上4个点。

解题：

第一题网上有答案

第二题：

这道题，思路1：

2. 首先，4 * 4的格子，放9个点，有 $C(16, 9) = 11440$ 种
3. 暴力，把这11440种里面，符合要求的找出来
4. 对于给定的矩阵，和符合要求的对比，看看哪一种移动次数最小

理论上，这就是 $O(1)$ 的时间复杂度，因为时间复杂度是一个常数，和输入数据没有关系

思路2：

9个点，最好情况就是，不用移动就符合要求，

最差情况就是要移动其中7个点(因为 $16 - 9 = 7$ ，只有7个空位了)

所以直接暴力，枚举要移动哪些点，到哪些目的地

时间复杂度也是 $O(1)$ 的

网易游戏

- 项目相关：项目亮点、印象深刻
- 技术选型
- 项目层面的优化
- 安卓和IOS兼容性，举个例子说说
- webpack的打包构建流程
- webpack的4要点（entry、output、loader、plugins）
- webpack打包比较慢怎么办
- 输入URL发生的过程
- 浏览器缓存
- CDN的原理
- flex布局？常用来干啥？常见的属性；grid布局？
- 浏览器刷新频率？60赫兹？requestAnimationFrame的题目——根本没用过啊喂
- JS事件循环机制 + 八股文输入输出题（process.nextTick todo）
- HTTPS为什么能保证安全
- HTTPS一定是安全的吗？
- 说一下中间人攻击——todo没说出来
- Vue的数据双向绑定原理
- 如何处理线上故障
- 编程题

// 1题 leetcode 20题

/*

2,将array 数组（仅包含整数的无序的数组）进行重新排列，
排序后数组符合特征 $array[0] \leq array[1] \geq array[2] \leq array[3] \dots$ 。

示例:

输入: array = [2,7,2,1,6,3]

输出: [2,7,1,6,2,3] 或其他符合要求的答案

```
1 function sort(array = []) {  
2     if (!array || !array.length) return []  
3     if (!array.length === 1) return []  
4     array.sort((a, b) => a - b) // 快排  
5     let i = 1  
6     while (i < array.length) {  
7         swap(array, i, i + 1)  
8         i += 2  
9     }  
10    return array  
11 }  
12 function swap(arr, i, j) {  
13     let temp = arr[i]  
14     arr[i] = arr[j]  
15     arr[j] = temp  
16 }  
17 const test = sort([1, 2, 3, 4, 5])  
18 console.log(test) // [ 1, 3, 2, 5, 4 ]
```

二面 46min 都是HR类型的问题，说好是业务面啊喂

- 前东家的情况巴拉巴拉
- 印象最深刻的项目？
- 回首过去，那些事情可以做得更好？
- 印象最深刻的bug？
- 项目优化的地方有哪些
- 职业规划已经期望薪酬
- 感觉都像闲聊的话题。。。

三面 HR面

HR很nice，没有什么坑

阿里文娱

1. 项目详情和难度
2. ocpc和ocpm的区别

汇量科技

1. 树结构设计和遍历
2. 模块联邦实现
3. 项目相关 (ocpc和ocpm)
4. 如何实现表单和表格模块化
5. 如何做一个小游戏

网易云音乐

一面：

1. 自我介绍一下你比较得意的项目，你做了哪些事，结果怎样样（项目巨无细致的挖）
2. http 缓存，内存缓存和磁盘缓存的区别，哪些文件存在这两内存
3. 你写过什么插件
4. vue key 的作用
5. 其他忘了

二面：

1. 项目相关
2. 手写防抖
3. 其他忘了

腾讯一面

先给半小时写三道题，有调试系统

1. 实现一个函数判断版本号大小， $0.1 < 1.1.1 < 1.8 < 15.1$ ，大于返回1，小于返回-1，等于返回0
2. 给定一个数组，找出里面和为target的组合，不可重复利用，且复杂度为 $O(n)$ 。如
`find([2,2,1,1,1,5,4,3,7,8], 3)`得出`[[2, 1],[2, 1]]`
3. 要求至多交换一次两个数位置，得到最大数字。如2763输出7263

- 4.讲解下上面的题，一些细节优化，实现对应优化。
- 5.自我介绍和最有成就的项目，项目细节，根据项目会问为啥不采取哪种更优化的方式，如果让你按这种优化思路是啥？
- 6.实现防抖，如果第一次要先调用呢怎么实现？
- 7.实现promise.all
- 8.实现继承
- 9.强缓存和协商缓存？具体字段和优缺点。为啥要有last - modified、etag两种，具体区别是啥？
- 10.http2.0特性？多路复用最后如何组装？首部怎么判断是否已经发过？
- 11.讲讲vue是怎样的一个框架？数据驱动实现？双向绑定原理。vue2的define Property存在哪些问题？具体哪些无法监听。vue3的实现。
- 12.xss攻击的防范
- 13.template怎么解析成对象的？html怎么渲染到页面，cssom和dom树具体怎么结合的？AST讲一下。
- 14.nextTick实现原理，哪些场合会用到？为啥能确保拿到dom了？ui渲染在事件循环哪一步？
- 15.有啥问题想问的？

凡科笔试

1. 解释下css3的flexbox
2. 解释下box-sizing
3. css性能优化
4. this考察

```
function Foo() {  
  getName = function () {  
    console.log(1);  
  };  
  return this;  
}  
  
Foo.getName = function () {  
  console.log(2);  
};
```

```
};  
Foo.prototype.getName = function () {  
  console.log(3);  
};  
var getName = function () {  
  console.log(4);  
};  
function getName() {  
  console.log(5);  
}  
//请写出以下输出结果:  
Foo.getName(); // 2 直接调用静态方法  
getName(); // 4 函数提升后函数表达式覆盖声明的函数  
Foo().getName(); // 1  
getName(); // 1  
new Foo.getName(); // new (Foo.getName)() 2  
new Foo().getName(); // (new Foo()).getName() 3  
new new Foo().getName(); // new ((new Foo).getName)() 3
```

5. 手写深浅拷贝

6. 手写reduce

7. 手写promise（注意篇幅，只有一面空间，我写个constructor就几乎快没空间了。。。）

8. 手写call、bind、apply

9. 浏览器渲染过程

10. 手写深度优先与广度优先遍历