一面：

1. CSS 栅格布局
2. CSS 伪类和伪元素
3. 根据条件获取递归树中过的某一节点
4. JavaScript this 的指向；箭头函数的 this 指向
5. Promise / setTimeout 的执行顺序；实际考察知识点：对「事件队列 / 宏任务 / 微任务」的了解
1. 协商缓存
2. 打开谷歌浏览器会有多少进程
1. 根据自己简历和做过的项目，问一系列相关问题。
2. 闭包的输出值，考查闭包（看试题给结果）
3. 浏览器缓存的方法有哪些，它们的优先级是怎样的 ？
4. 状态码 304 是什么意思 ？
5. 都说要减少 https 的请求，https 为什么慢 ？
6. http2 与 http1 有什么区别
7. click DOM 节点的 inner 与 outer 的执行机制，考查事件冒泡与事件捕获 （看试题给结果）
8. for 循环中的 var 、let 与 const 区别，for(const i = 0; i< 3; i++){ console.log(i); } 会输出什么结果 ？（看试题给结果）
9. 有没有系统学习过 es6 或者看过 es6 的书 ？
10. js 单线程、宏任务与微任务的执行顺序 （看试题给结果）
11. 考查箭头函数的 this 与 普通函数的区别，this 的指向 （看试题给结果）
12. vue 中 computed 与 watch 的内在是如何实现的 ？
13. 接下来前端要深入的方向 ？

算法：

- 写一个方法输出 ABCDEFG 的值（看试题、现场写程序）
- 从排好序的两个链表中，找到相同的节点，并输出链表（看试题、现场写程序）

1.事件循环
2.react diff 算法，key 的作用，setData 的机制，事件合成
3.vue 的 v-model 原理
4.实现一个方法，参数是一个 generator 函数，执行结果是执行完所有 generator 中的 yield
5.获取页面所有 img 并且下载
6.两个同源 tab 之间的交互，数据同步
1. 冒泡算法
2. 前端安全 ， DOS
3. 前端缓存、回话机制
4. 跨域
5. 计算机网络知识 TCP UDP
6. 测试 单测、集成测试
7. 自动化集成
8. Docker 应用
9. Nodejs express koa
1、换行字符串格式化
2、屏幕占满和未占满的情况下，使 footer 固定在底部，尽量多种方法。

3、日期转化为 2 小时前，1 分钟前等
4、多个 bind 连接后输出的值
5、原码，补码，反码
6、事件委托
1.throttle 实现（限流设计）
2.css 三行省略
3.vue 双向绑定
4.浏览器 reflow 和 repaint
5.webpack 性能优化
6.css3 新特性
7.proxy
1.宽是高的一半的垂直居中，里面有字体也要垂直居中类数组
2.promise async set time out 先后次序
3.event 类 on once 灯方法
4.==的隐式转化
1.浅拷贝与深拷贝区别
2.描述一个深拷贝流程，涉及循环引用
3.setTimeout 动画与 CSS animation 区别
4.CSS 硬件加速
5.JS 事件循环
6.改写一段代码，不考虑随机延时的影响，按顺序输出一段数字
7.设计一套机制，两个千位大数相乘
vue 的 $nextTick;计算属性如何更新;display 的各个属性值 ;盒模型的两种计算方式 ;前端跨域 ;jsonp 的实现（要代码）方式；后台响应头要加什么；代码实现连续点击 3 次每次间隔不超过 1 秒然后弹出页面（类似安卓的关于手机彩蛋）；给说有数组对象添加一个方法，返回出现频率>=n 的元素列表；
一面就主要围绕项目，底层构建，项目优化，工具链解决方案，框架实现基本原理，数据中间切层，数据结构与算法
1.自己实现 bind 函数
2 什么是闭包
3 最长子序列
4.二叉树中序遍历
5 .http 握手原理
6.react 新版本的特性
1、多空格字符串格式化为数组
2、bind 函数运行结果
3、点击 table 的 td 显示 td 内容
4、数字千分位处理
5、固定日期与当前时间格式化处理
6、上中下三栏布局
7、实现一个子类实例可以继承父类的所有方法
1、实现 sum(1)(2)(3).valueOf()，实现这么一个 sum 函数，返回 6
2.taskSum(1000,()=>{console.log(1)}).task(1200,()=>{console.log(2)}).task(1300,()=>{console.log(3)})，这里等待 1s，打印 1，之后等待 1.2s，打印 2，之后打印 1.3s，打印 3
1. Jsonp 跨域

2.js 原型继承 & 原型链
3.promise
4.二叉树搜寻算法
5.算法：前端做并发请求控制
1.节流函数
2.Koa 中间件机制及代码实现
3.React Fiber 原理以及为什么 componentWillRecievedProps 会废弃
4.给定一个数组，一个期望值，找到数组中两个相加等于期望值
1. html meta 标签有啥作用
2. cookie 结构有什么字段
3. bfc 块级格式化上下文
4. css <link/>为什么要放在头部
5. react 生命周期
6. react diff 算法
7. react 合成事件原理
8. http 请求都包含哪些字段
9. http 请求幂等性
10. versions 是一个项目的版本号列表，因多人维护，不规则
var versions=['1.45.0','1.5','6','3.3.3.3.3.3.3']
要求从小到大排序，注意'1.45'比'1.5'大
sorted=['1.5','1.45.0','3.3.3.3.3.3','6']
1. 数组去重
2. React Hook 原理
3. 列表 diff 中 key 的作用
4. Vue v-model 原理
5. 场景题：Vue CheckBoxGroup/CheckBox 设计
6. Vue 双向绑定原理
1. 实现 debounce
2. 实现 mvvm
3. react fiber 原理
4. react vdom 和 vue 的区别
1. React Hook, Fiber Reconciler ,新的生命周期
2. getDerivedStateFromPros 为什么是 Static
3. redux 异步
4. redux 异步中间件原理
5. express koa 中间件原理
1. 宏任务微任务
2. libUA
3. express ctx 中间键代码实现
4. vue 发布订阅和虚拟 dom 代码实现
5. 请实现如下的函数，可以批量请求数据，所有的 URL 地址在 urls 参数中，同时可以通过 max 参数控制请求的并发度，当所有请求结束之后，需要执行 callback 回调函数。发请求的函数可以直接使用 fetch 即可
1. 二叉树遍历

2. 并发请求最大值是 10，怎么处理队列
3. css 画出一个三角形
4. node 网关
5. csrf/xss 攻击原理
6. react diff 原理
css：
水平垂直居中
双栏固定布局
javascript；
1. 用 es5 模拟实现 function bind 方法
2. 模拟实现 截流 函数
3. 指定数组 找出 数字之和为 n 的组合 eg:
Arr [1 , 2 , 3 , 4].
n 5
Return [1, 4]
4. 组合继承
5. 实现 当 ul 点击时 输出 子元素 li 的内容
1. 怎么将一个异步方法 promise 化，以及实现 promise.all()方法
2. vue 单页多页的区别，vue 路由实现原理
3. vue 数据驱动视图原理？更新视图的过程是否是同步的操作？
4. nodejs 相关的应用（答：开发命令行工具、web 服务，ssr，数据库操作等）
5. vue 项目开发环境如何配置？wepack-dev-server 热更新功能实现原理
6. express、koa、redis 等技术相关应用
7. [1,2,3].map(parseInt) 执行结果
1. 40 分钟聊简历项目
2. 函数并发任务控制
3. 二叉树是否存在某个路径的和等于一个值
1. function request(urls, maxNumber, callback) 要求编写函数实现，根据 urls 数组内的 url 地址进行并发网络请求，最大并发数 maxNumber,当所有请求完毕后调用 callback 函数(已知请求网络的方法可以使用 fetch api)
2. throttle 函数实现
3.requestAnimationFrame 和 setTime、setInterval 的区别，requestAnimationFrame 可以做什么
4.二叉树路径总和（leetcode 112）
5. 给定一个不含重复数字的数组 arr,指定个数 n,目标和 sum,判断是否含有由 n 个不同数字相加得到 sum 的情况（leetcode 40 变种， 数字不得重复使用）
1、输入一个日期 返回几秒前 几天前或者几月前；
2、153812.7 转化 153,812.7；
3、用两种方法 一种是正则；
4、还有关于 bind 的一道题；
1. 写一个 eventBus
2. 元素水平垂直居中
3. vuex mobox
4. 小程序架构优化
5. 日志系统

1、Bind 方法手写实现
2、手写代码二叉树深度为 n 的遍历，遍历有哪几种方式
3、promise.then 的调用
4、promise.all()的实现原理
5、div 的点击事件回调不执行的原因，具体的一种原因怎么定位问题
6、hybrid 实现 bridge 的方法
7、最有挑战的项目
8、小程序框架的实现原理
1、实现 EventBus
2、 Vue 的组件通信
3、 Vuex 的响应式原理
4、 Webpack 优化
5、 http 缓存
6、 vue/react 代码复用的方式
7、 React hook 的应用场景
8、 Redux/Vuex 区别
9、 Vuex action/mutation 区别
10、css line-height

二面：

1. 聊了过往的项目经历，询问具体的技术方案和细节实现
2. 分割字符串；实际考察知识点：对「正则表达式」的了解
3. 富文本编辑器的实现；
4. 文件上传的实现；
5. 网络安全：CSRF & XSS 是什么及防范措施
6. 同源策略；跨域的实现方式
7. 带超时，带防重名的 JSONP 的实现

主要是围绕你的项目经历和技术，有一定的深度，主要还是要对项目全面熟悉；还有一个就是函数柯理化的编码实现
二面算法（85%）异步的问题、看结果补全程序、围棋模型检索类算法+设计模式（15%）单例模式、工厂、适配器、装饰器、订阅者-发布者、观察者 实际使用场景和实现
上来直接让写一个 autocomplete 组件，可能是想考察业务思考点；
后续的问题主要会接着业务场景问 扣实际场景 不问知识理论；
http 网络协议 ；
tcp 为什么是可靠的；
js 设计模式；
solid 原则；
1、curry 函数实现；
2、https 原理
3、webpack 打包原理
4、babel 原理

5、node 相关基础问题
1. css 单行和多行截断
2. 给一个由域名组成的字符串进行按子域名分组的反转，比如 news.toutiao.com 反转成 com.toutiao.news 需要 in place 做
3.其他技术问题都是穿插在我的业务项目里面的，有点针对实际情景给解决方案
1、实现一个 outsideclick 的 Hoc，触发时调用 子组件的 outsideclick 方法
2、手写一个 redux middleware
1. 最近在做项目（痛点，难点，怎么解决）
2. ssr（ssr csr 混合怎么处理）
3. 小程序架构（带来的优缺点）
4. 状态管理
5. 异步编程（各个优缺点）

三面：

1. 聊了过往的项目经历，询问具体的技术方案和细节实现
2. 在现团队中担任的角色；觉得自己印象最深刻 / 最有亮点的项目经历
不是技术细节了，更多的是大方向思考技术以及技术深入研究
1.自己做得最有成就的项目
2.自己主动承担并是核心的项目
3.项目深度:比如现场实现 vue 的数据代理等
4.技术广度:什么是微前端等
5.职业发展
1. js 实现依赖注入
2. 接口攻击的方式和防御措施
3. https 握手过程
4. 设计模式
5. redux 和 mobx 的区别
6. js 多线程如何共享大的数据
1. 小程序架构优化，
2.二叉树
3. diff 算法
4. 页面渲染原理
5. 图像算法
6. 事件循环
7. 长列表渲染
8. 前端安全

笔试：

虚拟 dom 深度递归算法实现原理
dom react 原理
css 布局
js 原型链继承
fetch 取消

eventloop
instanceof
promise 封装 setstate
redux 基本组成和设计单向数据流
vuex 底层流程和实现原理以及数据流向
https 协议的过程
http 的方法有哪几种
https 获取加密密钥的过程
类式继承的方案
三个继承方式的优缺点
prototype 继承的实现
实现一个 bind 函数
odejs 的事件循环
千位加逗号
数字千分位处理，正则和非正则都要实现
借用构造继承，几种组合继承方式
看编程代码说出运行结果：
Process.nextTick，setImmediate 和 promise.then 的优先级
Process.nextTick，pronise, setImmediate 的优先级