

2022年4月前端面经

笔者上一段工作在某大厂工作约3年，遇到了一系列困惑和成长阻塞。所以不得不换了份工作（在之前的工作一直拿到较好的绩效，并非被排挤、pua等）

工作期间比较疑惑的几个问题

- **p8\p7 应该是什么样的能力？**（我看到的周围高T大多数技术方面很水，抓住了风口跟随公司成长了上来，不否认格局、认知很牛，但是从技术的角度来看对团队几乎没有任何正向作用）。这会给我们带来恶性循环：「我们看不到优秀的大佬应该是什么样的，自身不能良性成长」
- **学习技术是不是只能依赖github or 刷题。** 笔者花了半年多时间去研究vue的源代码，并且成功提了一些pr、后面也陆续给其他知名开源项目写了一些pr、跟风刷 `type-challenge`，目前可以无压力通关。**but**发现自己并不能创造价值（我写了这么多代码，那我能不能创造某个工具）最开始自认为基本上可以。后面发现了肯定的答案：**不能**。当你看过了evan、antfu等一系列大牛后才知道自己的战斗力多弱。
- **试着找领导沟通** 团队在做什么技术产品？低代码（PS: 场景匹配低且实现效果一般）、组件库（PS: 有自嗨倾向且难度系数低）、rn探索（emm?...）

综上所述，如果你遇到了同样的问题，那么你一定不要犹豫，不要浪费时间，抓紧run。

某乎 *emm...*

埋点做过吗，如何实现的

跨域是怎么处理的

跨域方法有哪些

如果浏览不支持cors该怎么做呢？（这个我还真不知道，说了通过jsonp\nginx解决？）

ssr有了解过吗

绑定事件最后一个参数是什么？源事件了解过吗*（高难度问题）*

代码题：

// 1.使用 *React* 类组件或函数组件实现一个登录表单，要求：

// 含有一个名为 *username* 的输入框，它的值会随着用户输入实时更新

// 含有一个名为 *password* 的输入框，它的值会随着用户输入实时更新

// 含有一个提交按钮，点击此按钮后在控制台打印出已填写的用户名和密码值

代码题：

2.使用原生 *JS* 实现一个屏幕居中的弹窗。

Click

要求：

点击此按钮后屏幕中央出现一个弹窗

此弹窗水平和垂直方向都居中，弹窗内的文本内容是按钮的文字，即“**Click**”。

弹窗出现3秒后自动消失

不建议、感觉技术方面emmm

so app

vue3有哪些优化

性能优化你了解哪些

聊了大约20分钟，面试官人很好，但不推荐

小红书

一面（1.1h） 超级nice的面试官，交流了一些关于vue的思考

你做过的最重要的项目

移动端性能优化措施

vue为什么需要注册为component，而react不需要，就能直接使用（组件大小写）

什么是csrf。csrf的解决方式有哪些

低代码实现方案

vuex的缺点

react的理解有多少

ts支持

为什么vue的组件需要注册component才能使用，react怎么做到不需要注册也可使用

js 复制代码

写代码

快排

二面 (1.1h)

部门负责人

js 复制代码

```
[[1,2],[2,3],[4]]
```

判断两个点之间的路径有多少种，以及最短距离是多少

```
function(entry: [number, number]) {  
  return [  
    number, // 路径数量  
    number, // 最短距离  
  ]  
}
```

我写的：dfs + 记忆化搜索

项目相关的问题..

离职的原因，然后还有就是你期望的环境是什么样的（这个问题回答的不好）

三面 (0.7d)

大团队负责人

之前做过的项目有哪些。

一些细节的实现方法等

感受：可考虑，三面留下了差的印象。感觉不太nice的大老板。但是一面 + 二面都不错

某虾

这个小可爱的问题深度无语

一面 (1.5h)

缓存你用过的哪些，怎么用的？

localStorage你是怎么使用的，具体点，带实际经历场景等说**（不管是啥都要问你具体具体是怎么用的） **

http1、http2、http3的区别

url全链路过程

一道promise流程题，没记录下来

动画方案哪些，你是怎么用的

vue在2和3里，执行 `<div v-if="text"><div> text =false;` 全流程

postmessage你是怎么用的

service-worker你是怎么用的

cookie你是咋怎么用的

nginx缓存你是怎么用的，用哪个字段、怎么配置

跨域你是怎么解决的，分别用哪个字段，怎么配置

二面

给你一个promise队列，分别捕获在前1、2、3、4、5...n个promise都结束的结果

比如

```
excute([promise1, promise2, promise3]).then((data) => {
  console.log(data)
})
// 前N个都结束啦
// 前3个都结束啦
// 前2个都结束啦
// 前1个都结束啦
```

```
main().catch((e) => {
  console.log('top', e) // top networkerror
})

async function main() {
  try {
    loadImage() // uncaught promise
    loadConfig()
  } catch (e) {
    console.log('main', e) /// main logicbug
  }
}

function loadImage() {
  return new Promise((resolve, reject) => {
    setTimeout(reject, 1000, 'networkerror')
  })
}

async function loadConfig() {
  throw 'logicbug'
  await wait()
  console.log('configok')
}

function wait() {
  return new Promise((resolve, reject) => {
    setTimeout(resolve, 1000)
  })
}
```

某里 (1.1h)

一面

好像都是项目相关的问题，自我反馈出来的情况：

1、做项目的话需要深度

2、你对你期望的团队

二面

项目

业务，对业务的理解

三面

项目

monorepo的好处

新建一个活动、可提炼的能力有哪些

关于线上流程的稳定性

如何判断用户问题

整体不建议去：没有实际的代码问题、后面逐渐感觉到了pua的氛围

JERRY.AI(1h)

一面

js 复制代码

--

Problem Set below:

// Task: Implement a class named 'RangeList'

// A pair of integers define a range, for example: [1, 5). This range includes integers: 1, 2, 3, and 4.

// A range list is an aggregate of these ranges: [1, 5), [10, 11), [100, 201)

*/***

** NOTE: Feel free to add any extra member variables/functions you like.*

```

*/
class RangeList {
/**
 * Adds a range to the List
 * @param {Array<number>} range - Array of two integers that specify
beginning and end of range.
 */
add(range) {
// TODO: implement this
}
/**
 * Removes a range from the List
 * @param {Array<number>} range - Array of two integers that specify
beginning and end of range.
 */
remove(range) {
// TODO: implement this
}
/**
 * Prints out the List of ranges in the range List
 */
print() {
// TODO: implement this
}
}
// Example run
const r1 = new RangeList();
r1.add([1, 5]);
r1.print();
// Should display: [1, 5)
r1.add([10, 20]);
r1.print();
// Should display: [1, 5) [10, 20)
r1.add([20, 20]);
r1.print();
// Should display: [1, 5) [10, 20)
r1.add([20, 21]);
r1.print();
// Should display: [1, 5) [10, 21)
r1.add([2, 4]);
r1.print();
// Should display: [1, 5) [10, 21)
r1.add([3, 8]);
r1.print();
// Should display: [1, 8) [10, 21)
r1.remove([10, 10]);
r1.print();
// Should display: [1, 8) [10, 21)
r1.remove([10, 11]);

```

```

r1.print();
// Should display: [1, 8) [11, 21)
r1.remove([15, 17]);
r1.print();
// Should display: [1, 8) [11, 15) [17, 21)
r1.remove([3, 19]);
r1.print();
// Should display: [1, 3) [19, 21)

```

我的答复

js 复制代码

```

type IRange = [number, number];
type Mode = 'add' | 'remove';

// 把当前格式tuple转 array
type WriteableTuple<T> = (T extends Array<any> ? (T[number] extends readonly [infer A, infer B] ? [A, B] : T) : T);

class Rangelist {
  list: IRange[] = [];

  /**
   * @description: 与已有空间不存在交叉合并的情况
   * @param {[number, number]} input
   * @param {add | remove} mode
   * @return {null | number} idx
   */
  private filterInvalid = (input: IRange, mode: Mode = 'add') => {
    const [from, end] = input;
    if (input[0] === input[1]) {
      // 去掉相同输入
      return null;
    }
    let idx = this.list.findIndex((item) => item[1] >= from);
    if (!~idx) {
      // 最右边的case
      if (mode === 'add') {
        this.list.push([from, end]);
      }
      return null;
    }
    if (end < this.list[idx][0]) {
      if (mode === 'add') {
        // 直接插入的case
        this.list.splice(idx, 0, [...input]);
      }
      return null;
    }
  }
}

```



```

        return idx;
    };

    /**
     * @description: add调用, 让input区间合并到this.list[idx]上
     * @param {[number, number]} input
     * @param {number} idx
     * @return {void}
     */
    private merge = (input: IRange, idx: number) => {
        const idxRange = this.list[idx];
        idxRange[0] = Math.min(idxRange[0], input[0]);
        idxRange[1] = Math.max(idxRange[1], input[1]);
    };

    add = (range: IRange) => {
        // 找到第一个, 右小于等于from的; 后mixin
        let idx = this.filterInvalid(range);

        if (typeof idx === 'object') { // null
            return this;
        }
        const finalPoi = this.collect(range, idx);

        this.merge(range, idx);
        this.merge(this.list[finalPoi], idx);
        this.list.splice(idx + 1, finalPoi - idx); // 删除掉中间位置
        return this;
    };

    print = () => {
        console.log(JSON.stringify(this.list).replace(/\](?!$)/g, ' '));
        return this;
    };

    /**
     * @description: 从idx位开始收集, 找到所有的区间内数组
     * @return {number} 最后一位相关数组
     */
    private collect = (input: IRange, idx: number) => {
        const end = input[1];
        const len = this.list.length;
        for (let i = idx; i < len; i++) {
            const item = this.list[i];
            if (item[0] > end) {
                return i - 1;
            }
        }
        return len - 1;
    };

```

```
};
```

```
/**
```

```
 * @description: 同理于merge
```

```
 */
```

```
private detach = (input: IRange, idx: number): IRange[] => {  
    const [from, end] = input;  
  
    const appendRanges = ([  
        [  
            this.list[idx][0],  
            from  
        ], [  
            end,  
            this.list[idx][1]  
        ]  
    ] as const).filter(item => item[1] - item[0] > 0)  
  
    return appendRanges as WriteableTuple<typeof appendRanges>  
};
```

```
remove = (range: IRange) => {  
    let idx = this.filterInvalid(range, 'remove');  
    if (typeof idx === 'object') {  
        return this;  
    }  
    const finalPoi = this.collect(range, idx);  
  
    const appendRanges = (finalPoi === idx) ? this.detach(range, idx) : this.detach(range, idx).  
    this.list.splice(idx, finalPoi - idx + 1);  
    this.list.splice(idx, 0, ...appendRanges);  
    return this;  
};  
}
```

```
const l = new RangeList();
```

```
l.add([1, 5]);
```

```
l.add([10, 20]);
```

```
l.add([20, 20]); // [1, 5) [10, 20)
```

```
l.add([20, 21]); // [1, 5) [10, 21)
```

```
l.add([2, 4]); // [1, 5) [10, 21)
```

```

1.add([3, 7]); // [1, 7) [10, 21)

// l.add([8, 10]); // [[1,7),[7,21)]

1.remove([10, 10]); // [1, 7) [10, 21)

1.remove([10, 11]); // [1, 7) [11, 21)

1.remove([15, 17]); // [1, 7) [11, 15) [17, 21)

1.remove([3, 19]); // [1, 3) [19, 21)

1.print();

```

二面

代码题 40分钟2道M+的题 。比较challenge 大概28分钟做完，剩下10分钟做一些优化

这个题没给出最优解，「最优解基于位运算」

```

// 给定两个整数，被除数 dividend 和除数 divisor。将两数相除，要求不使用乘法、除法和 mod 运算符。
// 返回被除数 dividend 除以除数 divisor 得到的商。
// 整数除法的结果应当截去（truncate）其小数部分，例如：truncate(8.345) = 8 以及 truncate(-2.7335) = -2
// 输入：dividend = 10, divisor = 3
// 输出：3
// 解释：10/3 = truncate(3.33333...) = truncate(3) = 3

```

js 复制代码

二分查找，给出了最优解

```

// 整数数组 nums 按升序排列，数组中的值 互不相同 。
// 在传递给函数之前，nums 在预先未知的某个下标 k ( $0 \leq k < \text{nums.length}$ ) 上进行了 旋转，使数组变为 [nums[k]
// 给你 旋转后 的数组 nums 和一个整数 target ，如果 nums 中存在这个目标值 target ，则返回它的下标，否则返回
// 示例：
// 输入：nums = [4,5,6,7,0,1,2], target = 0
// 输出：4

```

js 复制代码

第一题没给出最优解，面试官并没有难为我

三面

讲了一些项目的东西

然后场景题：设计一个微博列表滚动

- 1、前后端交互应该有哪些内容
- 2、分页时候，用户刷新了数据，后端应该怎么做保证不重复。
- 3、页面数据量大，滚动的一些方式。
- 4、不同的分页方案的差别是什么

其他的忘了？

四面

为什么Set的复杂度是 $O(1)$ ，如何做到的

treemap和hashmap的实现以及区别

http1、2、3区别。3中如何实现快速握手。

dns解析方式

还有一堆很难得好问题，忘了，gan！！！！

大概就是：很算法、很难

五面

cto面试，他基本都是英文，聊一些经历之类的。

总结：**强烈推荐**，我没去是因为自身想再国内卷一卷。每个面试官都超级无敌nice。最后一面是王国辉大佬，真大佬，那边纯js做前后端，应该有很多机会和挑战

微软

pre

1、改错题，忘了

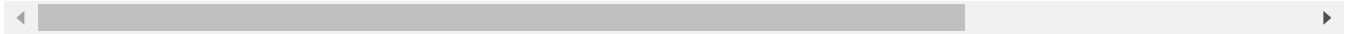
2、给一个数组，每一位是字符串，找出可以构成的无重复最长字符串['abc','cdf','pe']最长就是abcpe

3、忘了，大概是找出操作次数最少的，可以让字符串成为唯一字符的次数

4、给一个二进制的数字，'111000'。要么处以2要么减1，找到操作最少的次数。（这个题有性能测试，目前没找到有效

找到了解决方案，基于bigInit来做，`BigInt(2 ** 53).toString(2)`

5、找到给定字符串中最短的匹配字符串（要求如果有它的大写，就应该有它的小写）



一面 nice面试官

1、聊项目

2、网络的dns了解有多少

3、cdn了解有多少

4、dns解析过程

6、请求太多怎么优化、一个页面的img太多了有什么解决方案

7、埋点怎么控制show事件只执行一次

8、map和weakmap的区别

9、1px问题解决方式

10、gpu底层怎么工作的

代码：

已知fx递增，求它的反函数(调用fx，然后二分法。核心在于分析递增区间)

二面

项目相关，然后介绍了angular和react的一些数据管理手段

代码题:

判断是不是搜索二叉树（写出来了，但是严重被绕晕了，延伸到中序遍历相关的知识）

三面

ts的作用

范型

js 复制代码

实现ipv4的判断

js 复制代码

实现ipv6的判断

```
const replaceStart = (str) => {
  if (str === '0') {
    return true;
  }
  if (str.replace(/^0*(^0)/g, '') !== str) { // 这块直接reg.test就行，这么写麻烦了
    return false;
  }
  return true;
};

function checkIp6(str) {
  const arr = str.split(':');
  if (arr.length !== 8) {
    return false;
  }
  return arr.every(item => {
    if (item.length !== 4) {
      return false;
    }
    return item.split('').every(v => {
      const val = parseInt(v, 16);
      return val >= 0 && val <= 16;
    });
  });
}

'2001:0db8:85a3:0000:0000:8a2e:0370:7334';
console.log(
  checkIp6(
    '2001:0db8:85a3:0000:0000:8a2e:0370:7334'
  )
);
```

这个用ts体操来实现也算是个ts入门题

四面

js 复制代码

一些场景题：

有点记不住了

分享一些好东西：团队鼓励开源参与、团队做e2e和test、下班6点？

五面

忘了要面试，面试官等了我50分钟.....

1、项目为主

分享一些关于团队评估。

当场给了通过sde2，感恩。

总结：所有面试官+hr都是无敌nice，hr叫chufan，给的薪酬整体也很满意，总包较大，强烈推荐，和网上传言的差别很大

字节

一面 体验不佳

感觉面试官是个中级工程师，水平不高，更适合考察校招生

说一说从原生js到现在，你对前端发展趋势的理解

对其他语言的了解

学习方法

class 写声明函数和普通函数的区别（问的是class中的箭头函数的区别...）

js的执行机制（问的是预编译相关，没理解到）

还存在一些我感觉并不专业的「专业术语」，记不清了

1、寻找最长不重复字符串

没写最优解，大约1分钟完成。直接口述了最优解。因为很简单没必要深究

2、class Scheduler{}

这个题网上有很多，感觉并不难写，就不记录了

这个题描述有问题，没说返回，说的是输出，它的函数就是`console.log`。导致gap，然后写了3个版本的，最终我也不确定到

后来网上看了一下，其实这个题目本身还不错

3、实现promise\promise.all

promise简单写了一下，问题很多，面试官没要求去优化，比较好说话

promise.all 由于写过，3分钟输出

二面

怎么判断线上的手机白屏

离线包如何判断优先级

从前端视角去推动业务优化

如果觉得lottie-js的资源体积太大，如何去优化这个问题呢

首页怎么优化

性能优化的方案有哪些

动画的手段优缺点

js怎么实现的调用bridge流程知道吗

自动生成埋点代码怎么实现的

如何知道一个页面的访问成功率（进来一半就退了）

// ... 还有一些记不住了，但是都是特别好的问题，通过面试带来思考，起码时间没有浪费掉，大家感兴趣可以私信我，让这个面试官给你二面

// 需求看作包，转化为0-1背包问题

方案1、dp。结构大概如下：第0位表示做当前需求、第一位表示不做当前需求

```
[
  {
    acc: 0,
    user: [1,0,0]
  },
  {
    acc: 1,
    user: [1,2,0]
  }
]
```

方案2、贪心

按照需求不需要人数排序，比如一个需求需要人数为0，排到最前面，然后「优先支持需要人数少的需求」

三面

项目相关

假设某个项目想做公司平台要考虑哪些，以及怎么去做

react、vue、zepto等选型问题

react和vue性能对比。

埋点平台功能哪些

jsb、axios、fetch进行封装、支持实例、拦截等。写的并不好，因为太大了，但是题目很好

四面

全是业务

总结：二面体验最佳。整体体验不是最佳的（不如微软），但是这边会做很多深度复杂的业务。可以有机会学习到很多知识

重点来了

经过笔者入职2个月的观察，对当前的团队给予高度满意的评价。（确认了解团队，不想为了内推而坑人！！！极其厌恶为了内推无底线坑的人）

- 大家和部门领导日常交流完全对等（日常开领导玩笑等等）
- 深度跨端，和架构团队合作，比如：
 - lynx react第一个版本是我们这边写的
 - 直播容器能力开发
 - 前端做安卓、ios开发（不会有前端和客户端互相推责任的问题）
 - 比如webview预热、bridge重构等均是我们自己定制
 - nsr、prefetch、snapshot、ssg等
 - and so on
- 没部门没有小组群。只有一个部门群（40人+）
- 几个小组长都极其的好，并且技术上有很多创意的想法

来这边学习到了很多跨端的真正概念。也第一次去想rn为什么性能差（我一直以为rn很牛逼）。我的下一份工作基本不会再考虑国内大厂了。

- 北京: job.toutiao.com/s/2LBdRMS
- 杭州: job.toutiao.com/s/2Lk7pfc
- 字节跳动校招投递链接: jobs.toutiao.com/s/2LBLsPk

特别强调

部门的差异性比公司的差异性还大、我只是分享了我遇到的面试经历，不代表该公司其他部门。切勿无脑黑

推荐的公司

- jerry.ai 是一家美国外企。我觉得很有前景，技术水平应该也不错。

- ms的hr chufan真的很nice，大家可以联系她。职业生涯遇到的最佳体验都在微软这个部门。

其他感受

- 面试没遇到刚ts体操的问题
- 外企要多刷算法
- 面试阶段都觉得不match的千万不要考虑
- 优秀的人比你想象的更多