

CSIG 一面

丢算法题

栈实现队列

- 方法1：一个栈做buffer，一个栈做data
 - push：把 data 转移到 buffer，放入 num 到 data，再把 buffer 转回 data
 - pop：直接从 data 出
- 方法2：维护 data1 data2
 - push：放入 data1，如果 data2 空，把 data1 栈顶数据给 data2
 - pop：data2 出栈，此时如果 data2 空，从 data1 补一个数据

Topk：求最大k个数

- 方法1：维护最小堆，堆满判定堆顶和 num，num 大，替换堆顶，然后堆顶 sink，维护堆结构
- 方法2：快排，选定 position，维护右边的那一部分

求环入口

快慢指针求交点，从开头和交点出发，再求一次交点

判断n个点能否构成一个凸多边形

[凸多边形](#)的定义任意一边无限延长后其余边在这一边的同一侧

判断一个点是否在n多边形内（可凹）

从这个点做一条射线，计算它跟多边形边界的交点个数，如果交点个数为奇数，那么点在多边形内部，否则点在多边形外。

判断一个点在几个n多边形内

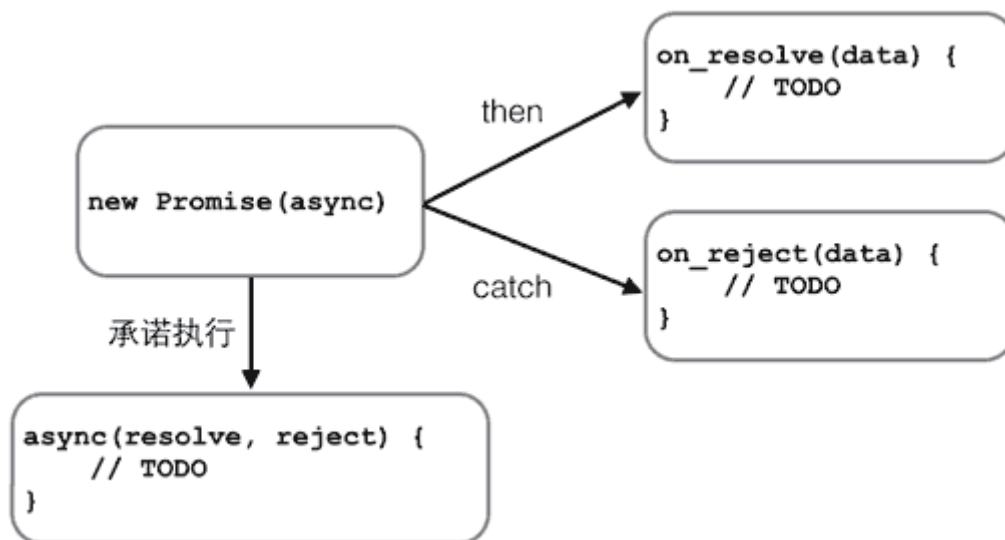
B树，找不到答案

JS

js如何处理异步

promise

Promise最大的好处是在异步执行的流程中，把执行代码和处理结果的代码清晰地分离了：



Promise还可以做更多的事情，比如，有若干个异步任务，需要先做任务1，如果成功后再做任务2，任何任务失败则不再继续并执行错误处理函数。

要串行执行这样的异步任务，不用Promise需要写一层一层的嵌套代码。有了Promise，我们只需要简单地写：

```
job1.then(job2).then(job3).catch(handleError);
```

追问：最新的？

- 原先：需要先获取 `code`，那就得嵌套，约束异步执行流程

```
function getFinal(){
  console.log("我是getFinal函数")
  getCode().then(function(res){
    if(res.data.code == 0){
      console.log(res.data.code);
      var params = {
        code: res.data.code
      }
      getList(params).then(function(res){
        if(res.data.code == 0){
          console.log(res.data.list);
        }
      })
    }
  })
}
```

getFinal();

- 现在： `async` 和 `await`

```
async function getResult(){
  console.log("我是getResult函数")
  //等待获取code
  let code = await getCode();
  console.log(code.data.code);
  if(code.data.code == 0){
    var params = {
      code: code.data.code
    }
```

```
    }  
    let list = await getList(params);  
    console.log(list.data.list);  
  }  
}  
getResult();
```

js都是单线程执行的？没回答出来

在JavaScript的世界中，所有代码都是单线程执行的。

由于这个“缺陷”，导致JavaScript的所有网络操作，浏览器事件，都必须是异步执行。异步执行可以用回调函数实现：

网络

从浏览器输入一个url，发生了什么

- 讲了好多，以后要注意

不考虑分包/四次挥手，一次发送报文耗时50，求发一个报文的时间

- 三次握手 + 回一次ack - 200
- 第三次握手的ack报文可以携带数据

数据库

为什么用B+树

- 提高读取数据时速率，减少磁盘IO次数

项目相关

问的不多

自由提问

- 部门？腾讯健康，用nodejs + cpp
- 你得转c++了！