**一、关于头条的面试小 tips**

**面试准备：**头条考工程的职位也很喜欢考算法题（甚至是数学微积分，比如手写红黑树插入

算法），需要认真复习基础的算法知识点

**训练技能：**多刷 leetcode acm 等题目，平时多看书总结

**面试形式：**1、视频面试 2、现场面试：现场手写代码

**面试轮次：**1 面-2 面-3 面-hr 面

**面试准备周期：**1-2 周为最好，准备时间不宜过长，在知识点吸收的最佳状态即可

**软性考察：**硬技术问完了，可能会考软技能（问你工作中最钦佩的人物是哪个或哪两个？为

什么钦佩他们呢？你对自己的职业规划如何？）

**注意事项：**leetcode acm等题目多刷，现场手写代码也会考察.硬技术问完了，可能会考软技能（问你工作中最钦佩的人物是哪个或哪两个？为什么钦佩他们呢？你对自己的职业规划如何？你具有geek精神么？你内心的渴望是什么？）

**二、头条职位要点**

基础架构类职位通关要点：底子好+熟悉底层基础知识+熟悉内部逻辑。

偏好人选：懂底层逻辑上手快+会思考（做业务擅长总结，花时间看书，把自己做的东西融

会贯通）

**职位理解：**就是你底子一定要好，底层的基础知识要清楚，无论做什么要懂内部逻辑。

服务端分为上 中 下

上层就主要做业务开发（可以理解就是产品需求，）相对要求简单些。

但是也是希望基础好一些，算法OK。因为我们的业务逻辑相对复杂一些。

基础架构我们目前招聘的就是中间层，主要给做业务开发的提供工具，节省他们大量的逻辑代码时间，也是要求较高的。纯底层就不理解业务也不太行。

所以，我们的综合要求就是：不管做啥岗位，基础要好。你要懂底层逻辑，你上手才快，而且要会思考的人，不能光做业务不总结。

会平时花时间看书，把自己做的东西融会贯通的人基本上是可以过的。

**面试套路：**平常用什么框架？为什么用这个？怎么用？

就是问问题问得越来越深 深到你回答不出来为止 然后综合判断你的技术深度和广度。不要犯低级错误，有思考能力就ok。会从基础知识 编码能力 算法和数据结构 工程能力 还有软素质这几块考察。软素质又包含了沟通推动能力 学习能力（聪明度） 产品意识 这些。

一般一面挂的基本就是基础和算法，然后就是项目说不清楚（沟通不行！）一面过了之后后面过的概率比较大，基本上卡一面。hr会主动联系你 你们自己协商面试时间，如果你觉得你要时间准备可以找个理由往后推。按自己舒服和合适的时间来，没关系的。所以一定要一面过，如果觉得岗位不合适我们可以转岗到其他岗位。全集团操作的，但是一面如果说你基础不好，就惨了。一面考基础，基础到你大学的那些东西。

**三、题目汇总**

**算法题（考细节、考基础）**

1、红黑树(手写红黑树插入算法)

2、CNN 神经网络

3、mle 推线性回归

智商题：一个楼层摔东西的经典题（100层楼，就是哪个楼层瓶子刚好摔碎）

**工程题**

1、操作系统

2、链表、二叉树（二叉树的非递归遍历）

3、多线程（生产者消费者等问题）

4、分布式训练细节

5、编程3sum

6、变种的二分查找

7、动态规划

8、操作系统是怎么实现共享内存的

9、虚拟内存的实现原理

**基础C++清单：**

1、C++为什么要有class

2、指针是什么？你能不用指针写C++程序么？指针是好还是不好

3、vector和list有什么区别

4、C++为什么要有类型

5、vector在resize时是原来的多少倍？

6、explicit 关键字的作用

7、dynamic\_cast等几种强制转换分别用在哪几种场合

8、使用过可变长模板么？使用在什么场合

9、如何解决多继承造成的类成员重复问题

10、类的static函数可以访问类的普通成员么？为什么？

11、用cpp11之前的版本实现反射，闭包机制，判断输入类的继承方式

12、cpp多态实现原理

13、容器rotate的三种实现

14、vector如何动态扩容？stl默认内存池如何实现

15、手写一个shared\_ptr（不考虑各种异常）

16、仅遍历一次ip字符串，判断该ip格式是否正确

17、cpp实现向量的延迟计算

18、cpp的内存定位泄漏和profile的工具

19、C++程序如何做单元测试

20、C++如何和其他语言交互

21、说说你对并行思想的理解及其实现方式，包括进程线程和集群，谈谈每一种的调度方式，稳定性，资源控制特点

22、如何看待第三方库？如果你想引用一个第三方库，程序里面要做哪些操作？文档里面要做哪些操作？若某一天因为某种原因，第三方库不能继续使用，如何更换

23、如何做程序优化？

程序里面有3种语言，C++ matlab python.运行在2台机器上（CS架构的影像服务器）。代码量比较大，svn代码目录有1gb多，构建一次工程需要2小时左右。

应该如何着手优化程序性能？

24、平时如何写文档？谈谈对设计文档和代码注释的理解，怎么看待团队统一代码风格？

25、说一说sizeof在计算变量所占空间大小时采取的机制

26、虚函数和虚继承对于一个类求sizeof的影响有什么差别

27、虚函数的实现机制

28、三次握手是通过什么方法来保证通信双方确认多正确

29、不同类型的变量如何与0做比较

30、如何判断一段程序是由C编译程序还是由C++编译程序编译的

**数据库和实现原理**

1.数据库myisam innodb区别

2.mysql索引实现原理

3.b树和b+树区别 复杂度

4.innodb中事务的实现原理

5.innodb中次级索引实现原理

6.redis数据类型，使用场景

7.redis里set 和zset 区别，以及两者的底层实现

8.(接上文)跳表里高度怎么定，map里的key和值都是什么类型，复杂度

Linux操作系统相关

9.什么是用户态和内核态，为什么要分这两个，以及这两者来回切换会耗时，有什么优化案例。

10.页面置换算法 FIFO LRU OPT

11.虚拟地址和物理地址是什么，为什么要有这两个

**网络**

12.tcp和udp区别，都有哪些典型应用场景

13.tcp有哪些机制保证可靠性

14.超时重试如果时间间隔设置的比较大会有什么问题

15.tcp三次握手四次挥手，为什么四次

16.为什么客户端最后要2个msl等待

**其他**

17.select poll epoll 区别

18.epoll性能高在哪里

19.红黑树

20.水平触发 边缘触发

**算法**

21.复杂度为o (n logn)的排序算法

22.快排复杂度证明

23.找数组第二大数

24.单链表排序 优化

考察：基础知识面，深度 ，编程能力。

2018.12.20基础知识：

1. Tars：是否可以超过使用阶段，了解内部实现

2. ping基于什么协议：

3. 创建一个线程的开销：

4.mutex和spin\_lock的区别：

5.golang底层的goroutine调度：

6.golang的slice底层实现原理：

7.redis：是否可以超过使用阶段，大致知道实现内部数据结构，说出细节

8.git：能使用，对实现原理是否清楚

9.Http包完整性检测方法：

10.长链接和短链接的区别： 长链接底层如何维护

11. python多线程机制：

2018.12.25编程题目：

1. 给定一个字符串，要求找出字符串的子串，子串中的字符不重复，且子串为所有子串中最长

2. 给定一个非负整数数组，你最初位于数组的第一个位置。数组中的每个元素代表你在该位置可以跳跃的最大长度。判断你是否能够到达最后一个位置

2018.12.26面试题目：

随机森林与gbdt的区别

knn中k取值的影响

kmeans迭代过程

数据倾斜的处理

多个字符串（可能多余2个）求最大公共字串

2018.12.29面试题目新增：lru的实现

怎么快速发现热门文章（以便缓存起来防止大量重复请求文章详情）

你们的rpc框架怎么做服务发现；registry节点挂了怎么办

一个线上游戏要给用户推荐附近的10个用户，已知经纬度，如何快速计算返回

redis有没有用过

mysql如何保证数据一致性（比如一条数据有多个线程会读取和更新）

tcp三次握手

泛洪攻击如何应对

2019.03.04更新题目

寻找峰值 n>=3, a[0] < a[1], a[n-2] > a[n-1]，相邻元素不相等，找出其中一个m，使得a[m-1] < a[m] && a[m] > a[m+1]

m\*n的表格，从左上走到右下，每次只能往右或者往下走一格，有几种不同的走法

其他：

聚集索与非聚集索引

innodb隔离级别

对于可重复读的隔离级别，在select \* for update 如何加锁，有主健与无主健

select/epoll

map为什么用红黑术而不用平衡二叉树

除了树什么样的数据结构查询时间复杂度为logn

linux 内梳2.6的调度策略是什么

字符串转浮点

ip地址检验

2019.03.06面试题目：

挂链表

冲突再散列

hash冲突怎么解决

c++hash table怎么设计，还有锁之类的

分布式事务，mysql索引之类的

用Google搜一下host crowding，会出现题目。

算法题目

sum(1,2).value() // 3

sum(1,2)(3,4).value() // 10

sum(1)(2)(3)(4).value() // 10