阿里云开发者社区

阿里云技术面试红宝书

阿里云开发者社区Offer 5000 独家资料（一）

阿里云技术面试真题公开

云原生 | 大数据 | IOT | 数据库
等领域的30道面试真题

阿里云开发者社区

阿里云开发者社区

Offer5000一键投递简历

阿里技术微信公众号

阿里云开发者社区

阿里云开发者“藏经阁”
海量免费电子书下载

目录

面试题

/ 常见的 DNS 记录类型有哪些，常见的 DNS 攻击方式有哪些以及怎么防护?

/ 在进行 CDN 流量调度时，要考虑哪些因素?

扫码投递简历

流量产品事业部

我们以遍布全球的数千个强劲 IDC 节点和数以亿记的的内外部客户端为基础，和网络进行亲密接触，寻找同路英雄共创未来!

招聘职位：CDN 点播研发专家等 5 个职位

面试题 001

常见的 DNS 记录类型有哪些，常见的 DNS 攻击方式有哪些以及怎么防护

参考答案：常见的 DNS 记录类型包括但不限于 A，AAAA，CNAME，NS，TXT，MX，SOA 等。常见的 DNS 攻击包括但不限于：泛域名攻击，放大攻击，反射攻击等，常见的防护方式包括但不限于：提升程序性能 (比如 dpdk 的 DNS)，zone 限速，来源 IP 限速，来源 IP 白名单，响应限速等。

面试题 002

在进行 CDN 流量调度时，要考虑哪些因素?

参考答案：需要考虑的因素包括但不限于：业务特性、质量要求、命中率、资源冗余度、APP 端适配条件、客户端网络环境等方面。

面试题

/ 如何把 DDD 应用到实际项目中来?

/ 如何设计一个百万级 TPS 分布式系统架构?

新金融事业部

阿里云新金融事业部由阿里云智能和蚂蚁集团的金融业务合并而成，全面拉通阿里巴巴数字经济提金融科技和金融服务的能力。阿里云的云计算技术体系叠加蚂蚁集团的金融级平台技术和创新的 Fintech 技术，必将引领金融市场技术平台的革命浪潮，同时为客户创造更多新的业务价值。

面试题 003

请举例说明，你是如何把 DDD 应用到实际项目中来的。例子中需要包含具体的领域模型设计，这么做的理由，以及因为这个设计而引进的坑。

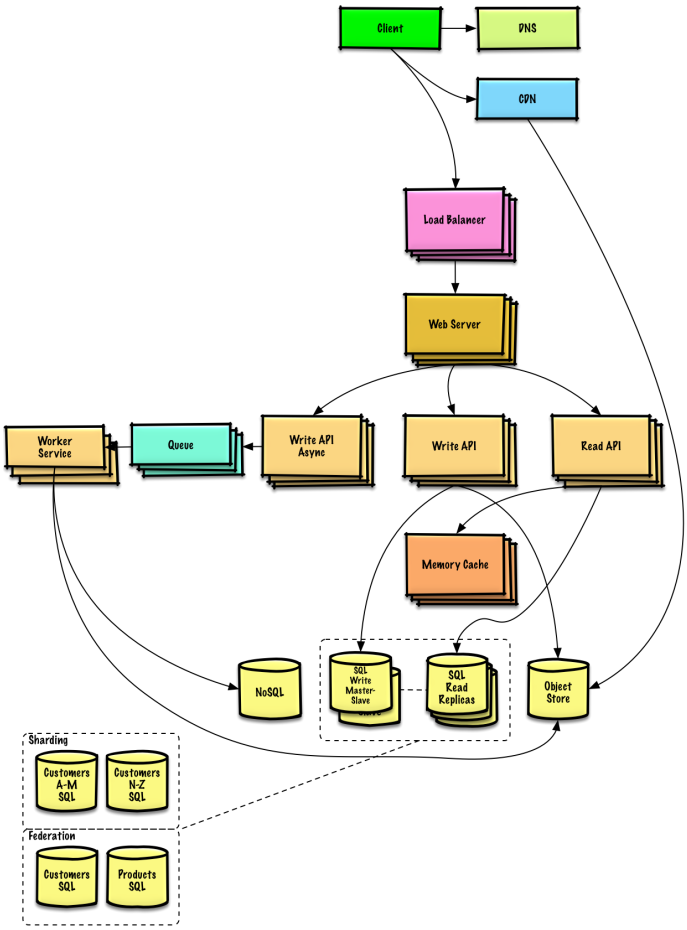
答案：本题为开放性问题，没有标准答案。

面试题 004

如何设计一个百万级 TPS 分布式系统架构？并举例说明这类分布式系统会引入哪些问题，以及对应的解决方案。

考察点：能够熟练应用缓存，消息中间件，数据库分库分表，读写分离等技术。

参考架构：



本图出自：https://github.com/donnemartin/system-design-primer

面试题

/ Java 多线程的协同

/ 数据结构的组合使用

扫码投递简历

计算平台事业部

在这里，你将和 Dataworks 云产品团队一起，整合阿里云海量异构存储元数据，统一各计算引擎调度，构建元数据中台，提供业界领先的大数据综合治理，数据湖，数据服务的能力。

招聘职位：java 开发等 4 个职位

面试题 005

（JAVA）有 3 个独立的线程，一个只会输出 A，一个只会输出 L，一个只会输出 I。
在三个线程同时启动的情况下，请用合理的方式让他们按顺序打印 ALIALI。
三个线程开始正常输出后，主线程若检测到用户任意的输入则停止三个打印线程的工作，整体退出。

考察点：

多线程的协同。

面试题 006

一个网站有很多页面（ url), 做一个 url 排行榜功能。排行根据 url 的访问次数 (pv) 排行。
排行榜需要实时准确即：某个页面每一次访问都会实时地影响到排行数据。
提示：排行榜本身也会有很高的实时访问需求，注意读和写的时间复杂度。

考察点：

数据结构的组合使用。

面试题

/ 异步、多线程并发控制如何操作?

/ MapReduce 思想、排序算法结合实例如何运用?

扫码投递简历

计算平台事业部

阿里云实时计算部是阿里巴巴大数据技术体系的核心团队。我们围绕 Apache Flink 为核心打造的大数据实时计算平台，一方面在阿里集团内部提供全公司范围的实时数据分析服务，支持淘宝、天猫、高德、优酷、饿了么、钉钉等所有阿里经济体子公司；另一方面通过阿里云向外界提供基于 Flink 的实时计算云产品服务，服务广大中小企业。

招聘职位：大数据计算平台研发专家等 3 个职位

面试题 007

题目案例

已知一个业务查询操作涉及 3 个 RPC 服务调用：query1, query2, query3, 其中 query1 耗时约 1 秒，query2 耗时约 0.5 秒，query3 耗时约 0.6 秒，且 query3 查询条件依赖 query2 的查询结果，

请编写代码，使该业务查询总体耗时最小。

考察点 1：异步

考察点 2：多线程并发控制

解题思路：

本题比较简单，主要考察知识点异步和多线程控制。

如果采用串行执行，query1+query2+query3 总耗时为 2.1 秒。

采用多线程异步并行执行，使用线程 A 请求 query1，同时使用线程 B 请求 query2 后再请求 query3（query3 依赖 query2 结果只能串行执行），

这样总耗时是 min(1, 0.5+0.6) = 1.1 秒。 需要熟悉 join, CountDownLatch 等线程协调控制方法，如果考生使用线程池则更佳。

面试题 008

淘宝 web 服务器上有 1 个 access 日志文件，记录着用户访问的 url，url 总数 100 亿以上，每个 url 约占 64 字节，

这些 url 可能存在重复，在一个内存只有 2G 的机器上，统计出访问频率最高的前 100 个 URL。

考察点 1：MapReduce 思想，利用中间文件存储，分而治之。

考察点 2：排序算法

解题思路：100 亿 *64/1024/1024/1024 = 596G, 可考虑分成 1000 个文件处理，每个文件大约 600M。

顺序读取文件，每行按照 hash(url)%1000 的结果将 url 写入到 1000 个文件中，这个过程是 mapreduce 中的 map。

针对每个小文件，使用 hashmap 统计每个 url 出现的次数，并使用堆排序得到访问次数最高的前 100 个 url，

将每个文件排序好的 100 个 url 及对应的 count 输出到 1000 个文件，最后将这个 1000 个文件 (此时每个文件只有 100 行) 进行合并排序。

面试题

/ 如何设计一个红包系统来满足春节期间千亿级红包的收发问题?

/ 如何选择最优索引?

扫码投递简历

数据库产品事业部

阿里云智能事业群数据库产品事业部承载着阿里巴巴经济体所有的在线数据处理分析以及数据库服务，是业务蓬勃发展的核心基础设施，在阿里云上全面覆盖电商、物流、安防、交通、健康、出行、气象、游戏、教育等广大行业并提供核心服务，在集团内为淘宝、天猫、大文娱等事业群提供稳定、高效、安全、低成本、极致性能与规模的数据库产品和服务。

核心产品包括：自研的新一代云原生数据库 POLARDB，新一代 OLAP 数据库 AnalyticDB，时序数据库 TSDB，以及关系型数据库服务 RDS、非关系型数据库服务 NoSQL，数据传输产品 DTS 和数据库大脑 CloudDBA。

招聘职位：数据库 OLAP 分布式计算研发专家等 5 个职位

面试题 009

面试题 1（难度：中等）：

如何设计一个红包系统来满足春节期间千亿级红包的收发问题？

本题为开放性问题，无标准答案，主要可以从以下几个不同的点对不同背景的面试人员进行考察：

业务：面试人员是否对抢红包的业务场景的需求有足够的了解，如果没有的话，是否有足够的沟通能力从面试官那里获取需要满足的业务场景的信息（比如金额方面不能有任何差错；比如对于并发量的要求等等）

底层存储：需要存哪些数据？用什么样的存储系统 / 数据库来落盘数据？数据存储的格式是怎么样的？数据库的 schema 怎么设计？底层存储如何拆分、如何架构导致其可以处理高并发请求？

缓存：缓存应该如何设计来提高系统的性能？如果使用分布式缓存？

同步 v.s. 异步：所有的操作都需要同步执行吗？有没有操作可以异步执行来减少对于用户的响应延迟？如果有的话怎么实现异步操作（比如消息队列）。

高可用：上述列出的组件之一如果宕机了怎么办？缓存被击穿了怎么办？怎么保证系统的高可用？系统怎么 failover？

测试与部署：如何压测整个系统？系统如何灰度？如何上线？

面试题 010

有如下表：

create table t(a int, b int, c int);

已知如下三条是这个表最常用的三条 query:

select * from t where a = 1 and b = 1;

select * from t where b = 1;

select * from t where b = 1 order by c desc;

以下索引哪个是最优的：

A. idx(a, b)

B. idx(b, a)

C. idx(b, c)

D. idx(a, b, c)

答案是 B

面试题

/ 举一个数据库快照读的实现方案

/ B+ 树和 B 树的区别有哪些?

扫码投递简历

数据库产品事业部

阿里云智能事业群数据库产品事业部承载着阿里巴巴经济体所有的在线数据处理分析以及数据库服务，是业务蓬勃发展的核心基础设施，在阿里云上全面覆盖电商、物流、安防、交通、健康、出行、气象、游戏、教育等广大行业并提供核心服务，在集团内为淘宝、天猫、大文娱等事业群提供稳定、高效、安全、低成本、极致性能与规模的数据库产品和服务。

核心产品包括：自研的新一代云原生数据库 POLARDB，新一代 OLAP 数据库 AnalyticDB，时序数据库 TSDB，以及关系型数据库服务 RDS、非关系型数据库服务 NoSQL，数据传输产品 DTS 和数据库大脑 CloudDBA。

招聘职位：MySQL 云数据库内核开发专家等 5 个职位

面试题 011

举一个数据库快照读的实现方案

参考答案：本题考察候选人对于基于时间戳的常见快照读的实现和一般 MVCC 机制的理解

面试题 012

B+ 树和 B 树的区别有哪些?

参考答案：本题考察候选人对数据库中经典数据结构的细节掌握度，以及常见的数据库扫描方法的理解程度

面试题

/ 编程实现 DAG（有向无环图）的 DeepCopy

/ 设计一个抽奖，假定只有非常有限的内存，如何处理一个无限的样本流？

基础产品事业部

扫码投递简历

阿里云智能云原生应用平台以容器和 K8s 为突破口，以分布式、微服务、服务治理、服务网格、消息、PaaS 为切入点布局产品技术，服务好阿里集团的同时，面向行业客户承担加速企业数字化转型升级，推动核心技术互联网化演进的责任，帮助企业客户和开发者全面拥抱云计算、最大化发挥云计算的价值、享受云计算的红利。面向未来定义研发、运维模式，推动 Serverless、函数计算等现代化架构演进，形成充分的产品技术竞争力，服务百万开发者，成为云原生时代的引领者。

招聘职位：云原生研发 Leader 等 4 个职位

面试题 013

编程实现 DAG（有向无环图）的 DeepCopy

这是一道编程题目，结合了数据结构和简单算法（如递归等）。

考察点：

1. 候选人应该明确什么是 DeepCopy 并主动沟通

2. 候选人应该清楚的定义数据结构

这个问题里面，需要定义节点，节点只要有 {value， Collection neighbors} 就可以了，增加别的成员一般是不合理的。

候选人应该意识到需要定义数据结构，如果不清楚定义（是个扣分项）需要提醒候选人。

图的话可以不定义单独的数据结构，有 Collection 所有节点集合就可以了。当然专门定义也可以。

有的候选人会有 Collection 和 Collection 定义（Node 里面没有 edge），或者有的候选人用二维链接矩阵，这也 OK。

数据结构定义合理性检查：

例如包含一些算法需要的 mutable variable 在数据结构里面，破坏结构定义封装以及 immutability 和 thread safety 的话，对于有经验的候选人是个比较大的减分项，对于学生一般是 OK 的。

3. 编程实现

一般来说比较方便的是用递归 /DFS 实现，候选人也可以选择其他算法（如 BFS）

以 Java 为例：

```
public static Collection<Node> DeepCopy(Collection<Node>  nodes) {
    Collection<Node> newNodes = new ArrayList<>();
    Map<Node, Node> copyMap = new HashMap<>();
    for (Node node : nodes) {
        newNodes.add(copyNode(node, copyMap));
    }
    return newNodes;
}

private Node copyNode(Node node, Map<Node, Node> copyMap) {
    if (copyMap.containsKey(node)) {
       return copyMap.get(node);  // 可以加上 nullability check
    }
    Collection<Node> newNeighbors = new ArrayList<>();
    for (Node neighbor : node.getNeighbors()) {
        newNeighbors.add(copyNode(neighbor, copyMap));
    }
   Node newNode = new Node(node.getValue(), newNeighbors);
   copyMap.put(node, newNode);
   return newNode;
}
```

4. 其他

这个编程的过程，经常出现的是递归的方法和 wrapper 方法之间划分不清，出现大量的重复代码，候选人这里花多少时间解决也是一个能力的表现。

还有经常有候选人意识不到 DeepCopy 里面需要保持图的结构因此想不到用 Map，这个也是不行的。当然如果要求不高，可以直接把题目编程 DeepCopy 一个树，这样就没有去重的需求了。

能够正确完成的，可以 follow up：如线程安全，问一下候选人方法是否是线程安全的（如果在 Node 节点里面存一些临时变量，或者把 Map 作为全局变量等就不是了），可以问如何改造成线程安全之类的问题。

另外 Follow up Big(O)：时间复杂度（O（V) + O(E))

面试题 014

问题：设计一个抽奖，假定只有非常有限的内存，处理一个无限的样本流，任何时候都能提供出 N 个中奖的样本，而且所有已经遇到的样本的几率一样

该问题比较适合：(1) 需要考察设计的；(2) 有一定的统计的背景，即便不了解 reservoir sampling 也可以自己去推理。

如果不熟悉这个算法没关系，候选人完全可以通过自己的思路去推演这个设计，不要求做出来，只是系统设计的思路。

面试题

/ 如果 Java 程序 CPU 飚高到 100%，怎么排查原因?

/ 多租户隔离是什么? 解决哪些问题? 多租户的架构是怎么样的?

IoT 事业部

扫码投递简历

IoT 的使命是数字化物理世界，从而建立基于数字孪生体的数字世界 (CPS)，并且在数字世界建立操作系统来调度物理世界，比如通过调度人货场；来提升新零售的效率，通过调度人机料法环来提升产业的效率。

招聘职位：前端开发专家等 5 个职位

面试题 015

如果 Java 程序 CPU 飚高到 100%，怎么排查原因?

考察点：系统运维的能力，常见命令的使用。

思路：Top -H 查看 CPU 飚高的线程，jstack 看看线程所在的 Stack 来分析问题。

面试题 016

多租户隔离是什么? 解决哪些问题? 多租户的架构是怎么样的?

考察点：面向企业场景的安全、高可用能力

思路：多租户分为逻辑隔离和物理隔离，解决的是企业数据安全和高可用的问题，可以从分布式切片的角度和统一路由的角度来回答该问题。

面试题

/ 图的基础概念以及基础数据结构。

/ 请实现代码片段，把地址 0x80008000 处的 32 位数据的第 10 位开始的 7 位设置为 0xAA。

扫码投递简历

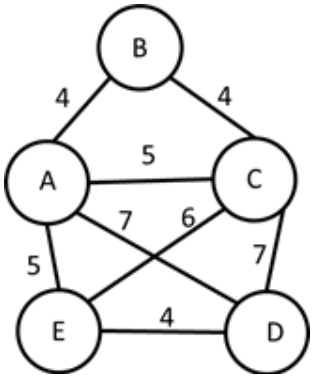
达摩院

平头哥主要针对下一代云端一体芯片新型架构开发数据中心和嵌入式 IoT 芯片产品，从云和端两个方面进行软硬深度协同的技术创新，目标是让数据和计算更普惠，持续拓展数据技术的边界。

招聘职位：编译研发专家等 5 个职位

面试题 017

已知有 5 个城市，现需要在城市间搭建通信电缆，各个城市间的造价如下，求最小造价。



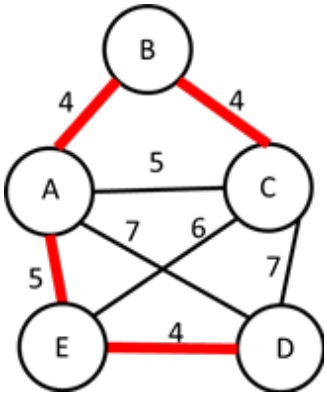
考察点：

图的基础概念以及基础数据结构

得到最小生成树的经典方法

参考答案：

Prim’s algorithm 普林演算法（贪心算法）：



C++ 实现：

```cpp
void prim() {
  priority_queue<ii, vector<ii>, greater<ii>> pq;
  for (auto p : e[0]) {
    pq.push(p);
  }
  visited[0] = true;
  int cost = 0;
  while (!pq.empty()) {
    auto [w, u] = pq.top();
    pq.pop();
    if (!visited[u]) {
      visited[u] = true;
      cost += w;
      for (auto p : e[u]) {
        if (!visited[p.second])
          pq.push(p);
      }
    }
  }
  cout << "MST's cost is " << cost << endl;
}
```

面试题 018

请实现代码片段，把地址 0x80008000 处的 32 位数据的第 10 位开始的 7 位设置为 0xAA。

考察点：

1. 对 C 的位运数的理解

2. 对绝对地址访问的理解

答案：

```
unsigned int tmp;

tmp = *(volatile unsigned int *)0x80008000;

tmp = (tmp & (~(0xff<<10))) | (0xAA << 10);

*(volatile unsigned int *)0x80008000 = tmp;
```

面试题

/ 设计和编码实现一个具备 LRU 过期策略的缓存程序。

/ 一个云端应用部署会涉及到哪几类配置信息?

扫码投递简历

市场营销 & 公共事务部

阿里云云开发平台是面向开发者和中小企业打造的一站式、全云端的开发平台，打开浏览器就可以开发、调试、上线，所测即所得，重新定义云时代原生的开发体验。

招聘职位：前端架构师等 2 个职位

面试题 019

考察对 java 基础能力，对 HashMap 熟悉程度：

设计和编码实现一个具备 LRU 过期策略的缓存程序。

答案：

```
public class LRUMap<K,V> extends LinkedHashMap<K,V>
{
    protected final int _maxEntries;
    public LRUMap( int maxEntries)
    {
            this(16, maxEntries);
    }
    public LRUMap(int initialEntries, int maxEntries)
    {
        super(initialEntries, 0.8f, true);
        _maxEntries = maxEntries;
    }
    @Override
    protected boolean removeEldestEntry(Map.Entry<K,V> eldest)
    {
        return size() > _maxEntries;
    }
}
```

面试题 020

一个云端应用部署会涉及到哪几类配置信息?

答案：1. 基础设施配置；2. 部署配置；3. BaaS 服务配

面试题

扫码投递简历

/ 高位特征的搜索，以寻找相似特征的个体，通常针对该问题的解决方法是啥？核心原理是啥？

/ 在目标跟踪算法中，若判定路径的统计学合理性。请列举几种可能的统计学模型来解决该问题。

全球技术服务部

阿里云智能全球技术服务部（Global Technology Services，简称 GTS）是一支面向云智能客户，提供完整生命周期服务的技术履约团队。旨在通过专业化的交付实施、系统化的中台支撑及标准化的服务产品能力，运用云和数据智能的力量，联合生态伙伴一起帮助客户实现业务价值。

招聘职位：业务中台交付架构师等 5 个职位

面试题 021

实际应用中经常需要我们对进行对某种高位特征的搜索，以寻找相似特征的个体。请问业界通常针对该问题的解决方法是啥，核心原理是啥。

参考答案：近似最邻近（ Approximate Nearest Neighbor）算法，基本分类：

基于树的方法

KD 树是其下的经典算法。一般而言，在空间维度比较低时，KD 树的查找性能还是比较高效的；但当空间维度较高时，该方法会退化为暴力枚举，性能较差，这时一般会采用下面的哈希方法或者矢量量化方法。

哈希方法

LSH(Locality-Sensitive Hashing) 是其下的代表算法。对于小数据集和中规模的数据集（几个 million- 几十个 million），基于 LSH 的方法的效果和性能都还不错。这方面有 2 个开源工具 FALCONN 和 NMSLIB。

矢量量化方法

矢量量化方法，即 vector quantization。在矢量量化编码中，关键是码本的建立和码字搜索算法。比如常见的聚类算法，就是一种矢量量化方法。而在相似搜索中，向量量化方法又以 PQ 方法最为典型。

对于大规模数据集（几百个 million 以上），基于矢量量化的方法是一个明智的选择，可以用用 Faiss 开源工具。

基于图的方法：

基于 k- 近邻图构造（k-neighbor-graph construction）。阿里跟浙大合作的算法 NSG，即为基于图的算法。

面试题 022

在目标跟踪算法中，需要判定路径的统计学合理性。请列举几种可能的统计学模型来解决该问题。

参考答案：一些经典模型如马尔可夫随机场（MRF），条件随机场（CRF），标签传播（label propagation）。

如果给出的答案没有提及明确的算法，但是接近于优化最大联合概率分布，也 ok。

面试题

/ 服务限流有哪些算法?

/ free -m 命令输出，buffer 和 cached 各是什么含义，有什么区别? -/+ buffers/cache 目的是看什么，分别是什么意思?

全球技术服务部

阿里云智能全球技术服务部（Global Technology Services，简称 GTS）是一支面向云智能客户，提供完整生命周期服务的技术履约团队。旨在通过专业化的交付实施、系统化的中台支撑及标准化的服务产品能力，运用云和数据智能的力量，联合生态伙伴一起帮助客户实现业务价值。

扫码投递简历

招聘职位：高可用技术专家等 3 个职位

面试题 023

服务限流有哪些算法?

参考答案：服务限流常见算法有并发计数器算法，漏桶算法，令牌桶算法。前两种算法不支持突发流量的限流，令牌桶算法支持突发流量的限流。 一般用谷歌 guava 落地令牌桶算法，用 sentinel 作为服务限流的中间件。

面试题 024

free -m 命令输出，buffer 和 cached 各是什么含义，有什么区别? -/+ buffers/cache 目的是看什么，分别是什么意思?

参考答案：buffer 是块设备的读写缓冲区，cache 作为 pagecache 的内存，文件系统的 cache。

查看内存真正被占用的容量（Used-buffers-cached），查看内存还可使用的容量（Free+buffers+cached）。

面试题

/ 模拟构造一个哈希表。

/ 如何用 GPU 来并行计算矩阵乘?

扫码投递简历

基础设施事业部

让基础架构 / 设施成为阿里的核心竞争力，做技术驱动的世界第一的商业基础设施服务商。

招聘职位：AI/HPC/BigData 大计算类基础设施高级产品专家等 4 个职位

面试题 025

模拟构造一个哈希表（哈希函数可以用 rand(seed) 代替），统计出哈希表在指定加载因子（实际元素数量除以哈希表容量）下 slot 被占用的比率、冲突链平均长度（用链表解决冲突）、最大冲突链长度。假设在查询哈希表时，一半元素命中，一半不命中，计算这种情况下的平均访存次数。

第一题答案。下面的代码并没有真正实现哈希表，每个 slot 位置只用了一个计数器来统计。答题者如果实现一个哈希表来真正模拟出结果也是可以的。

```
#include<stdio.h>
#include <stdlib.h>
#include <stdint.h>

#define  HASH_TBL_SZ  100000
int hsh_ary[HASH_TBL_SZ];

int  hash_stati( int ratio )
{
    int i, slots_occu = 0;
    int max_collision = 0;
    int  ele_nr = ratio * HASH_TBL_SZ / 100;
    double avg_col_len;

    if( ratio <=0 || ratio > 100 ){
        printf( "Error parameter: ratio must be between 1-100.\n" );
        return -1;
    }

    for( i = 0; i < HASH_TBL_SZ; ++i ){  // 也可用 memset() 等函数清零
        hsh_ary[i] = 0;
    }

    for( i = 0; i < ele_nr; ++i ){
        uint32_t idx = rand() % HASH_TBL_SZ;
        if( hsh_ary[idx] == 0 ){
            slots_occu ++;
        }
        hsh_ary[idx] ++;
        if( hsh_ary[idx] > max_collision ){
            max_collision = hsh_ary[idx];
        }
    }
```

```
    avg_col_len = (double)ele_nr/slots_occu;
    double slot_ratio = (double)slots_occu/HASH_TBL_SZ;
    printf( "slot occupied ratio =%f, avglen = %f, max_collision = %d\n",
slot_ratio, avg_col_len, max_collision );

    double access_nr;
    access_nr = 0.5 * avg_col_len + 0.5 * avg_col_len * slot_ratio;
    printf("average access number(50%% hit) = %f\n", access_nr);
}

int main(int argc, char**argv)
{
    hash_stati(10);
    hash_stati(20);
    hash_stati(50);
    hash_stati(80);
    hash_stati(100);
}
```

例子中 5 次调用的输出：

```
slot occupied ratio =0.095280, avglen = 1.049538, max_collision = 4
average access number(50% hit) = 0.574769
slot occupied ratio =0.181190, avglen = 1.103814, max_collision = 5
average access number(50% hit) = 0.651907
slot occupied ratio =0.392900, avglen = 1.272588, max_collision = 5
average access number(50% hit) = 0.886294
slot occupied ratio =0.551120, avglen = 1.451589, max_collision = 8
average access number(50% hit) = 1.125795
slot occupied ratio =0.632040, avglen = 1.582178, max_collision = 8
average access number(50% hit) = 1.291089
```

面试题 026

输入为：dimention 为 NxN 的数组 A，B，输出为矩阵 C；考察点就是如何用 GPU 来并行计算矩阵乘。每个线程算 C 里面的一个 element

第二题答案

```
#include <math.h>
#include <iostream>
#include "cuda_runtime.h"
#include "kernel.h"
#include <stdlib.h>

using namespace std;

__global__ void matrixMultiplicationKernel(float* A, float* B, float* C, int N) {

    int ROW = blockIdx.y*blockDim.y+threadIdx.y;
    int COL = blockIdx.x*blockDim.x+threadIdx.x;

    float tmpSum = 0;

    if (ROW < N && COL < N) {
        // each thread computes one element of the block sub-matrix
        for (int i = 0; i < N; i++) {
            tmpSum += A[ROW * N + i] * B[i * N + COL];
        }
    }
    C[ROW * N + COL] = tmpSum;
}

void matrixMultiplication(float *A, float *B, float *C, int N){

    // declare the number of blocks per grid and the number of threads per block
    // use 1 to 512 threads per block
    dim3 threadsPerBlock(N, N);
    dim3 blocksPerGrid(1, 1);
        if (N*N > 512){
            threadsPerBlock.x = 512;
            threadsPerBlock.y = 512;
            blocksPerGrid.x = ceil(double(N)/double(threadsPerBlock.x));
            blocksPerGrid.y = ceil(double(N)/double(threadsPerBlock.y));
        }

    matrixMultiplicationKernel<<<blocksPerGrid,threadsPerBlock>>>(A, B, C, N);
}
```

面试题

/ 如何使用优秀的时空复杂度快速找到这个数字?

/ 在两个有序自增数组中寻找第 k 大的数，并分析时间复杂度。

扫码投递简历

基础产品事业部

阿里云开放平台服务着百万级用户，拥有亿级访问量的平台与场景，是连接阿里云与客户的重要产品门户。在阿里云提出被集成理念后，开放平台已经成为阿里云最核心产品线之一，我们的使命是让企业更好地使用阿里云。为此我们需要在底层基础设施层面、企业服务能力层面、标准化体系层面、工具链编程体验层面等领域全面提升阿里云的能力，急需各界技术高手一起打造最先进的服务！如果你是技术 geek，或者有云计算深度使用经验，又或者有对企业服务领域有极大热情，欢迎加入我们！

招聘职位：开放平台工具研发专家 / 高级专家等 3 个职位

面试题 027

长度为 n 的数组，其中只有 2 个数字出现了奇数次，其他均出现偶数次，问如何使用优秀的时空复杂度快速找到这个数字

思路：主要是利用异或定理，若 a^b=x，则 a=b^x，b=x^a。

解法：

1) 假设这两个数分别为 a、b，假设将数组中所有元素全部异或后结果为 x，因为 a!=b，所以 x=a^b，且 x!=0

2) 判断 x 中某位 bit 为 1 的位置，例如 x=0x00101100，那么 k=2/k=3/k=5 皆可，只需一个；

3) 因为 x 中第 k 位为 1 表示 a 或 b 中有一个数的第 k 位也为 1，另一个数第 k 位为 0；假设 a 的第 k 位为 1；

4) 将 x 与数组中第 k 位为 1 的所有数字进行异或，最终结果得到 b；

5) 将 x 与 b 异或，得到的结果即为 a

面试题 028

在两个有序自增数组中寻找第 k 大的数，并分析时间复杂度。

思路：有多种解法，要求查找效率最高

解法 1：由于数组本身有顺序没必要全部再排序，可以定义两个游标分别指向两个有序数组，按序移动，并用 count 计数，当 count 等于 k 时，返回两个游标指向的数中较小的那一个，时间复杂度 O(m+n)，不是最优解

解法 2：充分利用两个数组都是有序的条件：

1）当 array1[k/2−1] == array2[k/2−1] 则返回 array1[k/2−1] 或者 array2[k/2−1]

2）当 array1[k/2−1] > array2[k/2−1] 则 array2 在 [0,k/2−1] 范围内的元素一定比 array1、array2 合并后第 k 个元素小，可以不用考虑 array2 在 [0,k/2−1] 范围内的元素

3）当 array1[k/2−1] < array2[k/2−1] 则 array1 在 [0,k/2−1] 范围内的元素一定比 array1、array2 合并后第 k 个元素小，可以不用考虑 array1 在 [0,k/2−1] 范围内的元素

因此算法可以写成一个递归的形式，递归结束的条件为：

1）array1[k/2−1] == array2[k/2−1] return array1[k/2−1]

2）array1 或者 array2 为空时，return array1[k−1] 或者 array2[k−1]

3）k==1 时，返回 min(array1[0],array2[0])

时间复杂度 O(log(m+n))

解法 3：归并排序，时间复杂度 O(nlogn)，此解法效率低，没有考虑实际情况，不得分；

面试题

/ 给定一个单向链表如何判断是否存在环，并给出环的起始节点

/ 连续子数组最大和

基础产品事业部

阿里云混合云团队责基础产品中的混合云存储产品，致力于研发混合云存储，分布式软件定义存储，灾备以及数据管理产品，依托遍布全球的阿里云基础设施，构筑安全，可靠，高效的混合云存储系统，助力企业客户数字化转型。

扫码投递简历

招聘职位：混合云存储管控开发专家等 3 个职位

面试题 029

给定一个单向链表如何判断是否存在环，并给出环的起始节点。

不存在环返回 NULL，存在返回环的起始点。

参考答案：

```
struct List
{
    int val;
    struct List *next;

}


List* FindStart(struct List *head)
{
      if (head == NULL || head->next == NULL)
    {
      return NULL;
    }

    struct List *slow = head;
    struct List *fast = head;
    struct List *meet = NULL;

    while (fast != NULL && fast->next != NULL)
    {
      slow = slow->next;
        fast = fast->next->next;

        if (slow == fast)
        {
            meet = fast;
            break;
        }
    }

    if (meet == NULL)
    {
        return NULL;
    }

    slow = head;
    fast = meet;
```

```
    while (slow != fast)
    {
        slow = slow->next;
        fast = fast->next;
    }

    return slow;
}
```

面试题 030

给定一个整数数组，有正有负有 0，比如 [7, 8, 10, −24, 15, 11]

1. 找出和最大的连续子数组的和

2. 找出和最大的连续子数组的起止 index

3. 找出最长的和最大的连续子数组的起止 index

考察能力：

1. 基础算法灵活运用能力

2. 初始值，数据类型，边界条件的严谨度

3. 随着题目变化不断深入的能力

参考答案

1. O（n）算法，从前往后叠加并记录当前和，当前和小于或者小于等于 0 时，从下一个数字开始重新叠加。注意：

   a) 全负的数组如何处理

   b) 记录和的变量的类型

c) 小于还是小于等于，差别在哪里?

2. 在发现更大的和的时候，更新临时起止 index

3. 核心是 1 中的小于或者小于等于 0，以及每次遇到一样的最大和时，判断长度

```
vector<int> getSum(vector<int>& nums) {
      int n = nums.size();
    int ans = nums[0], l = 0, r = 0;
    int cur = nums[0], tl;

    for (int i=1; i<n; ++i) {
        cur += nums[i];
        if (cur > ans) {
              ans = cur;
            l = tl;
              r = i;
        }
        if (cur < 0) {
              cur = 0;
            tl = i;
        }
      }

    return {ans, l, r};
}
```

阿里云开发者社区

Offer5000一键投递简历

阿里技术微信公众号

阿里云开发者社区

阿里云开发者“藏经阁”
海量免费电子书下载