**职位列表**

具体JD如下，欢迎各位广播。有靠谱简历，请速速转给denglu@baidu.com

**一、搜索核心研发工程师**

工作职责：  
-检索架构的设计与优化   
-数据挖掘与数据分析   
-推荐架构的设计与优化   
-音频内容分析与处理   
-用户行为分析，用户属性挖掘，用户关系分析和应用

职位要求：  
-精通C/C++或者PHP编程语言   
-对数据结构和算法设计具有深刻的理解   
-熟悉Linux/Unix系统、熟悉网络编程、多线程编程技术   
-计算机或相关专业本科以上学历   
-善于学习和运用新知识，具有良好的分析和解决问题能力   
-具有良好的团队合作精神和积极主动的沟通意识   
-具有以下条件者优先：有检索系统经验，熟悉数据挖掘，熟悉音频处理，熟悉分布式系统和Map Reduce等模型

**二、搜索实习研发工程师**

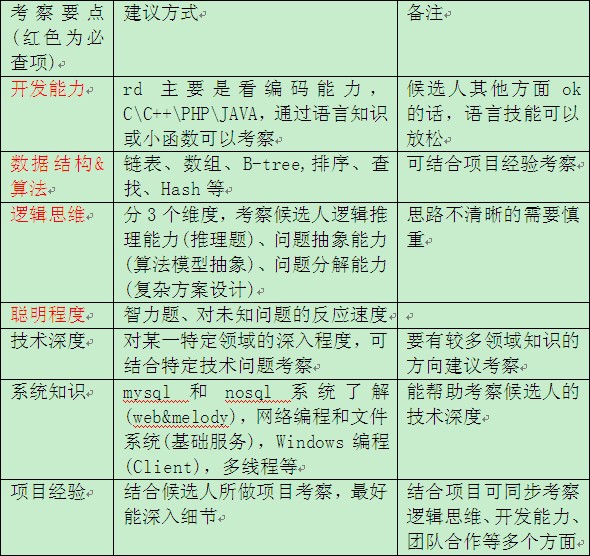
工作职责：  
-数据挖掘与数据分析   
-推荐架构的设计与优化   
-音频内容分析与处理   
-音乐基础曲库的设计与优化

职位要求：  
-熟悉C/C++或者php编程语言  
-善于学习和运用新知识，具有良好的分析和解决问题能力   
-具有良好的团队合作精神和积极主动的沟通意识   
-具有以下条件者优先：熟悉mysql的使用，有音频处理经验，熟悉数据挖掘等

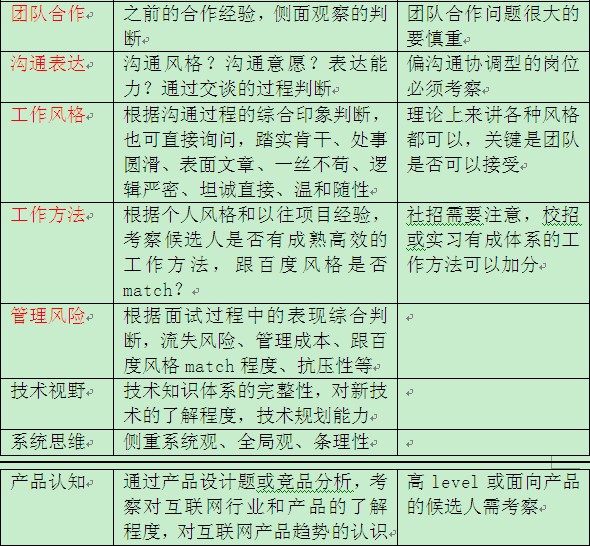
**面试流程**

这个流程主要针对rd工程师的面试而言，目前还处于试运行状态，各位面试官请参考执行。如有更好的建议欢迎反馈给denglu@baidu.com

**一、一面考察点**



**二、二面考察点**



**三、三面考察点**

1、有两篇还不错的面试经验：

<http://coolshell.cn/articles/4506.html>

<http://coolshell.cn/articles/4490.html> 从上面考察点也可以看出，一面重点考察经验和知识，二面和三面重点考察能力和潜力。

2、很多问题考察的时候，建议不要止于候选人“知其然”，更要考察是否“知其所以然”，这样能深挖出很多东西。

3、面试氛围的轻松自然是很重要的，可以让候选人更好的发挥，同时也更容易发现一些紧绷时不易流露的特征。

4、上面的分档不是绝对的。有一些考察点，如果前一位面试官没有考察到，建议后面面试官视重要程度补充考察。前面的面试官也可以视情况提前考察后面的点。

**面试题库**

这里摘抄的面试题来自于平时的网上收集，欢迎大家添砖加瓦。

**一、语言知识**

1、写一个strcpy函数，然后strncpy函数

2、int arr[12][20],请问sizeof(arr[4])是多大？

3、代码实现单链表逆序的函数？

4、这2套函数的区别？

1) fopen,fread,fwrite ; 2) open, read, write

(1)fopen 系列是标准的C库函数；open系列是 POSIX 定义的，是UNIX系统里的system call。

也就是说，fopen系列更具有可移植性；而open系列只能用在 POSIX 的操作系统上。

(2)使用fopen 系列函数时要定义一个指代文件的对象，被称为“文件句柄”（file handler），是一个结构体；而open系列使用的是一个被称为“文件描述符” （file descriptor）的int型整数。

(3)fopen 系列是级别较高的I/O，读写时使用缓冲；而open系列相对低层，更接近操作系统，读写时没有缓冲。由于能更多地与操作系统打交道，open系列可以访问更改一些fopen系列无法访问的信息，如查看文件的读写权限。这些额外的功能通常因系统而异。

(4)使用fopen系列函数需要"#include <sdtio.h>"；使用open系列函数需要"#include <fcntl.h>" ，链接时要之用libc（-lc）

小结：

总的来说，为了使程序获得更好的可移植性，未到非得使用一些fopen系列无法实现的功能的情况下，fopen系列是首选。

5、在main函数中调用return、exit、\_exit的区别？

(1)return()有返回的功能，一般用于函数的返回，且只能返回一级。该命令在main()主函数中时，也会使程序中断。

(2)exit()在stdio.h中，直接使进程停止运行，清除其使用的内存空间，并销毁其在内核中的各种数据结构，而且对于打开的文件（如 fread/fwrite，常常是读/写到内存缓存中的，一定规模后才一次性自动写入文件),回自动写回。一般有三种方式，exit(0)则进程返回值是 0，exit(1或-1)则进程结束返回值是1或-1，一般0默认为正常退出，且常用EXIT\_SUCCESSFUL代替（在matlab.h中定义为 0）。这样，程序员就可以知道退出的原因，以便于对程序进行调试。

(3)exit()在不同的头文件中，与exit()的区别是，在停止进程，清除内存空间的同时，不会写回和关闭流文件。如果有一些数据，我们认为已经写 入了文件，实际上因为没有满足特定的条件，它们还只是保存在缓冲区内，这时我们用\_exit()函数直接将进程关闭，缓冲区中的数据就会丢失，反之，如果 想保证数据的完整性，就一定要使用exit()函数。如IO文件流命令，fopen一个txt文件后，进行fprintf写入数据,但是，你这时打开该文 件，发现该文件是空的；然后知道你运行fclose之后，缓存里的数据才进入txt文件中。

6、calloc和malloc的区别？

1)主要的不同是malloc不初始化分配的内存，calloc初始化已分配的内存为0。

2)次要的不同是calloc返回的是一个数组，而malloc返回的是一个对象。

3、堆结构的相关问题

4、C、C++语言的参数传递方式

5、调用地址为0的子例程：

( **(void(**)())0)();

6、C++语言中，能否在成员函数中delete this？ 能，但调用之后不能在访问任何non-static成员，且该对象必须是new出来的

。。。

**二、数据结构**

1、用栈模拟队列？O(1)复杂度？

2、节点有重合的Y字形链表，如何检测？考虑有环的情况？

A：判断是否相交最简单的办法是遍历到尾，尾节点一样说明有相交。找到相交节点有多种方式：逐条遍历，用地址做hash判断是否到达相交点，只要遍历1条 半链表；先遍历两链表求得各自长度，假设为m和n，且m>n，那么对m长链表跳过m-n个节点，然后两个链表同时遍历，地址一样的那个点为相交点。

3、双向列表的插入删除

4、比较数组和链表的区别

5、把二叉树转化成链表有两个变种：一个是转化成单链表，一个是转化成双链表。

6、只给你一个指向链表中某元素的指针，请删除该元素。 将该节点后一个节点的值拷过来，然后修改直接删除后一个节点。注意，如果给出的是最后一个节点是删不掉的。

7、给你两个链表，如何判断其是否相交，如果相交，如何找到两个链表的第一个交点。

8、判断一棵二叉树T中，是否包含另一颗二叉树P的结构。

9、假如有一个很长的整数数组，要求用这个数组实现两个堆栈。 这个题目很简单，就是用两个栈指针，然后分别从数组的两头开始，向中间生长。具体写程序的时候，检查一下两个指针是否相遇就可以了。

10、如何用一个数组实现三个堆栈？ 答案其实是上面那个题目的一个挺巧妙的扩展：三个堆栈里面的两个，还是按照上面的办法做；然后第三个堆栈，从数组三分之一的地方开始，向三分之二的方向生 长。具体说来： 假设数组是A，长度是L，那么三个堆栈分别从下标0，L，和1/3L的地方开始生长。假设三个堆栈的栈顶指针分别是x，y和z，那么当栈生长到一定时间的 时候，空间会不够用，可能会有以下情况： x生长到了1/3L y和z相遇 这两种情况下，数组都会有一段空间空闲（如果三个栈一样长，那就结束了），或者是A[x，1/3L]，或者是A[y，z]。那么把这个空间继续按照上面的 办法分割，直到空间用完为止。 。。。

**三、算法**

1、A\B两个url集合，100w url，每条100字节长，怎么最快最省的找出重合的url并输出？使用字典的需要占用多少内存？如何解决冲突？

2、字符串集合A，其中有部分逆序重合的，如何最快的找出其中的逆序重合的部分。假如有100w字符串，每个字符串最常1024字节，那么需要占用多少空间？

A：最快的方法是用hash，通常会进一步问如何解决hash冲突，结合hash冲突的解决方法问占用时间和空间的问题。

3、给定一个包含4300000000个32位整数的顺序文件，找到一个至少出现两次的整数。

4、有一个文件，有很多很多的整数（也许有100亿），寻找其中最大的K个。

5、有一个很大的文件，存放一堆7位的电话号码，号码无重复，请用最小的内存消耗，将其排序。 6、给10MB的内存，给一个4百万整数的文件，找一个不在文件中的整数。

7、给你一个长度为N的整数数组，请找出最大的子数组和 8、有一个论坛，有ID发帖数目超过总数的一半，给你论坛所有帖子的ID列表，请你找到这个水王。

9、有一组数，除了一个数只有1个，其他都是两两成对的，请找出那一个不成对的数。另，如果不成对的数有两个，该如何是好。 10、有一个长度为N的链表，N未知。希望你只遍历一次链表，就从链表中等概率的挑出K个数。

11、设矩阵为A[m][n]现从A[0][0]进，从A[m-1][0]出，要求经历所有矩阵中的点，且不重复访问任何点，问总共有多少种路径？给出算 法思想，最好有程序实现 12、有一个01数组，现在要你对它进行重排（即保持0和1的数目不变），使得成为这样的形式：0101..01xxx.. 即01间隔，然后多出来的0或者1放到后面（用xx表示）。例如110110重排为010111。要求O(1)空间复杂度，以及只能遍历一次。

13、一个数组，其中只有一个数只出现一次，其余数皆出现偶数次。设计Time: O(n) Space: O(1)的算法得出那个只出现一次的数。用异或

14、如果只有两个数只出现一次呢？

--可以尝试对数组进行分割，将两个不同的数分到两组，然后异或即可。 将所有数全做一次异或，最终结果为a^b.因为a和b不等，那么a^b必然不为0。那么a^b这个数上面必然能够找到一个二进制位是1，在这个二进制位 上，a和b不等。 根据这个二进制位，将各元素中在这位上为1的分派到左边，为0的分派到右边，形成两个子数组。可以证明，1）这两个数组分别包含a和b。2）每个数组中除 了a或 b之外的所有元素都是成对出现的。

15、有两个数组A和B，假设都是整数类型的，而且都是从小到大排好序的，A里面，从下标0开始，有X个数；B里面有Y个数，A的长度是X+Y，B的长度是Y，就是说A里面能放下所有A和B的数。目标是设计一个算法，把所有这些数字都放到A里面去，当然结果要是排序的。

A：倒着做merge sort。具体说就是用两个下标，分别从A和B的数据尾巴开始，做合并。合并的结果从A的最后一个位置开始放，最后A里面就是A和B的合并的并且排序的结果了。

16、关于排序： 有什么方法可以不用比较就能排出顺序？

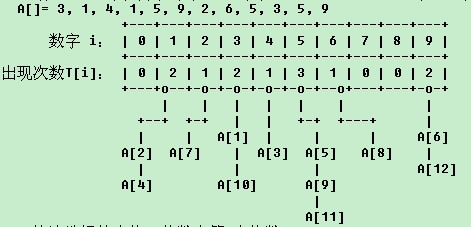
借助Hash表的思想，多数人都能想出这样一种排序算法来。 我们假设给出的数字都在一定范围中，那么我们就可以开一个范围相同的数组，记录这个数字是否出现过。由于数字有可能有重复，因此Hash表的概念需要扩 展，我们需要把数组类型改成整型，用来表示每个数出现的次数。 看这样一个例子，假如我们要对数列3 1 4 1 5 9 2 6 5 3 5 9进行排序。由于给定数字每一个都小于10，因此我们开一个0到9的整型数组T[i]，记录每一个数出现了几次。读到一个数字x，就把对应的T[x]加 一。

A[]= 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 9 +---+---+---+---+---+---+---+---+---+---+

数字 i： | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | +---+---+---+---+---+---+---+---+---+---+

出现次数T[i]： | 0 | 2 | 1 | 2 | 1 | 3 | 1 | 0 | 0 | 2 | +---+---+---+---+---+---+---+---+---+---+

但上述方法实际上并不是真正的排序，因为没有输出数值和原始位置的对应关系。用开散列的方法可以解决： 同样借助Hash表的思想，我们立即想到了类似于开散列的方法。我们用链表把属性值相同的元素串起来，挂在对应的T[i]上。每次读到一个数，在增加 T[i]的同时我们把这个元素放进T[i]延伸出去的链表里。这样，输出结果时我们可以方便地获得原数据中的所有属性值为i的元素。



17、快速选择给出的一些数中第k小的数？ 快速排序也有一个有趣的副产品：快速选择给出的一些数中第k小的数。一种简单的方法是使用上述任一种O(nlogn)的算法对这些数进行排序并返回排序后 数组的第k个元素。快速选择(Quick Select)算法可以在平均O(n)的时间完成这一操作。它的最坏情况同快速排序一样，也是O(n^2)。在每一次分割后，我们都可以知道比关键字小的 数有多少个，从而确定了关键字在所有数中是第几小的。我们假设关键字是第m小。如果k=m，那么我们就找到了答案——第k小元素即该关键字。否则，我们递 归地计算左边或者右边：当km时，我们递归地寻找右边的元素中第k-m小的数。由于我们不考虑所有的数的顺序，只需要递归其中的一边，因此复杂度大大降 低。复杂度平均线性，我们不再具体证了。

18、给定三个整数， 找出中间那个数， 要求比较次数尽量少，即使是最坏情况

19、假设这有一个各种字母组成的字符串，假设这还有另外一个字符串，而且这个字符串里的字母数相对少一些。从算法是讲，什么方法能最快的查出所有小字符串里的字母在大字符串里都有？

1）hashtable 2）我们可以给每个字母分配一个素数，从2开始。然后我们把大字串中的每个字母代表的素数相乘得出一个数，用小字串中的每个字母代表的素数去除它。如果除的过程中没有产生余数，则小字串是大字串的一个子集。 。。。

**四、设计能力**

1、设计一个缓存系统

2、实现一个字典，然后加可解决冲突？

3、有一blog系统，主要数据为用户、博客文章、博客评论，如何设计存储架构，能够方便的通过用户访问文章及其评论，同时也便于用户管理自己的评论？以及当用户量增长以后如何考虑数据切分？当出现热区用户时如何处理？

A：要能够同时方便的通过文章找到评论，以及通过用户找到评论，比较简便的办法是对于评论数据存两份，一份跟文章走，一份跟用户走。数据切分通常走垂直切分比较好扩展，但具体的处理方式要注意。通过定期统计，然后将热区用户数据独立出来，可以避免资源占用过高。

4、如何设计一个搜索引擎？大致分为哪几部分？哪块技术难度最高？

5、设计一个音乐盒系统？核心是那几块？如何做到可扩展？

6、如何实现一个读-写锁？--注意饥饿问题

7、一致性hash?

8、cap理论?base理论?

9、quorum NRW?

N: 复制的节点数，即一份数据被保存的份数。

R: 成功读操作的最小节点数，即每次读取成功需要的份数。

W: 成功写操作的最小节点数 ，即每次写成功需要的份数。

这三个因素决定了可用性，一致性和分区容错性

10、数据库的sharding?与partition有区别么?

11、对订票的12306系统，设计一个方案实现其中的余票查询模块， <http://timyang.net/management/design-review/>

12、针对facebook中的like功能设计一个方案

需求

* 可以对一个对象（一条feed、文章、或者url）进行like操作
* 需要看到一个对象的like总数
* 可以看到一个对象的like用户列表，优先显示我的好友列表(social list)。
* 数据量：每天新增的like对象数为1千万，每秒like计数器查询量及social user list均为30万次/秒。

假设现有系统的存储结构（MySQL）为

friends表, from\_uid + to\_uid 为联合主键

(from\_uid bigint,o\_uid bigint,ctime timestamp)

feed表, feed\_id + like\_uid 为联合主键

(feed\_id bigint,like\_uid bigint,ctime timestamp);

假设以上存储是单表结构，请问如何设计like技术方案？如何评判你的like方案的优劣？

13、如果你是linux内核工程师，针对8核CPU，怎么设计CPU复杂均衡算法？ <http://donghao.org/2010/07/kernel-smpiioouaoai.html>

**五、思维能力**

1、1-1000个数（无序），里面有2个缺失，如何找出缺的数？有2个重复呢？

2、求一个char（8bit）中，二进制1的个数，越快越好。

3、有一个整数数组A[N]，让你不用除法，求另一个数组B[N]，其中B[i] = A[0]\*A[1] ... \* A[N-1] / A[i]，期望复杂度是O(N)

4、给一组一维的空间 [1, 6] [2, 4] ... ，请求是否有区间重叠。

5、给一个整数N，求它的阶乘N!，有几个0结尾。

6、盒子里有三种颜色的球，红黄蓝，可以用任意两个不同颜色的球，换两个另外颜色的球，比如1红 + 1黄 = 2蓝。现在盒子里面有171个红球，172个黄球，173个蓝球，问，能不能经过若干次交换，最终变成同一颜色的球。

7、有一副扑克牌（你可以用任意方式来表示），被抽去一张，请快速找出这抽去的一张是什么？

有 1000瓶超级名贵的葡萄酒，其中有1瓶有毒。这种毒药很厉害，哪怕被稀释了1000000倍还是可以毒死人的但这个毒药一定时间后才会毒发，时长是 1个月。为了不浪费这些葡萄酒，有1000个壮士决定花5周的时间将毒酒找出，他们只希望最多有10个人牺牲，你需要如何安排才能实现

从纠错编码的表示来理解，1000个数据位，其中有一位是错的，不妨假设没有毒的用0表示，有毒的用1表示，则这1000个数据位中只有一位

是1。现在要用10个校验位来表示出错的位置，可以用汉明码，理论上是可以找到错误位置的，因为2^10>1000，符合汉明码的纠错条件。

另一种解法：

1)为1000桶酒编号，1~1000；

2)为每个人编号：A~J；

3)第一天，把酒分为10组，每份100桶，有1~100，101~200，···901~1000，然后每个人对应其中一份，即A对应于

1~100；

4)第二天，把酒取序号重新编组：如取原来每一组的前10个分为一组，即1~10，101~110，···901~910分为第一组。其它的类似，然后

让每个人又对应其中的一组。

5)第三天，把酒又重新编组：取（4）中所有分组中每个取出10个，即1、11、21、···991为一组，2、12、22、···992为一组，其它

的类似，然后让每个人又对应其中的一组。

6)这样，一个月后先死的一人就可以确定出哪一百桶有毒。

7)第二个死的人可以确定第一份里的哪10桶有毒。

8)第三个死的人可以确定10桶里哪一桶有毒。

8、假如我们班有n个MM，每一个MM都有一个独家八卦消息。两个MM可以通过电话联系，一通电话将使得双方都获知到对方目前已知的全部消息。要想所有n个MM都知道所有n条八卦消息，最少需要多少通电话？

一 个最简单的办法就是从n个人中选一个消息汇总人，所有n-1个人都打电话给她，她再打电话给所有人。这样总共需要2n-2通电话。其实，汇总阶段的最后一 通电话和发布阶段的第一通电话可以合并为一通电话，这样的话该方案实际上只需要2n-3通电话。如果n=4时，其实只需要4通电话即可。A<-& gt;B、C<->D之后每个人都掌握了2个消息，然后A<->C、B<->D即可。顺着这个思路，本题更优的答案 是2n-4.首先，选出4个人作为消息总人。其余每个人都选择一个汇总人并与之通话；然后4个汇总人再用4通电话互相更新一下消息（用刚才n=4的办 法）；最4个汇总人把电话再打回去，实现所有消息全部共享。 9、用哪四个相同的正整数（不限于 1 到 13 ）能算出 24 ？您能找出这个问题的所有 11 组解吗？（注意：这里算 24 的规则和平常一样，只允许加减乘除四则运算，可以添加括号改变运算顺序） 答案：这 11 个解分别是 3 、 4 、 5 、 6 、 12 、 22 、 23 、 24 、 25 、 26 、 48 ，您找全了吗？下面我们分别给出每组解的其中一种算法。有趣的是，多数人会理所当然地认为四个 24 显然能算出 24 ，随后却发现这并不是一件容易的事情。在继续看答案之前您不妨先自己想想该怎么算。 3 \* 3 \* 3 - 3 = 24 ，4 \* 4 + 4 + 4 = 24， 5 \* 5 - 5 / 5 = 24， 6 + 6 + 6 + 6 = 24， 12 + 12 + 12 - 12 = 24， 22 + (22 + 22) / 22 = 24， (23 \* 23 + 23) / 23 = 24 ，24 + (24 - 24) \* 24 = 24， (25 \* 25 - 25) / 25 = 24 ，26 - (26 + 26) / 26 = 24 ，48 / ((48 + 48) / 48) = 24

10、小 A 站在甲、乙两地之间的某个位置，他想乘坐出租车到乙地去。他看见一辆空车远远地从甲地驶来，而此时整条路上并没有别人与他争抢空车。我们假定车的行驶速度 和人的步行速度都是固定不变的，并且车速大于人速。为了更快地到达目的地，小 A 应该怎样做呢？你认为下面哪种思路是正确的？

(A) 由于车速大于人速，小 A 应该尽可能早地上车，充分利用汽车的速度优势。因此，小 A 应该迎着空车走上去，提前与车相遇。

(B) 为了尽早到达目的地，小 A 应该充分利用时间，马不停蹄地赶往目的地。因此，他应该自己先朝目的地走一段路，再让出租车载他走完剩下的路程。

答案：两种方案花费的时间是一样的。只要站在出租车的角度上想一想，问题就变得很显然了：不管小 A 在哪儿上车，出租车都要驶完甲地到乙地的全部路程，因此小 A 到达乙地的时间总等于出租车驶完全程的时间，加上途中接小 A 上车可能耽误的时间。

11、如何用火柴棒搭出一个正方形？可以任意多根，注意四边形本身是不稳定的结构，一定要先找出一个直角才行。

http://www.matrix67.com/blog/archives/3448

12、一个晚上，结伴同行的4个人要由一座小桥的这边到达另一边，过桥时必须用到手电筒，只有一支手电筒，而小桥每次最多只可以通过两人（小桥承受不了太多的重量），4个人的过桥速度分别为1分钟、2分钟、5分钟、10分钟，4个人全部安全通过小桥，最少需要多长时间？

如果加一个限制条件，将手电筒放在桥中间，朝左朝右各能照到桥的一半，最少时间是多少？

13、3分钟沙漏、7分钟沙漏，如何模拟9分钟？

14、2个盘子，10个蓝色球，10个红色球。

如何放置这20个球到2个盘子，使随机选个盘子，随机选个球，拿到红球的概率最大。概率为多少？

A：其中1个盘子放1个红球，其他球放另外一个盘子。概率为1/2+9/19

15、图书馆借书，还书后，突然某天收到书没还的消息。

如何确认问题在什么地方？

考察思考问题的逻辑性和完备性。

16、两个水桶。11升和17升，提供7升水，找出其中原理

最大公约数问题。

17、孪生素数问题，如果X和X+2都是素数，如何证明X+1能够被6整除？假如X>3。

A：X\X+1\X+2这3个数模3的余数肯定为0、1、2中某一个。因为X和X+2都为素数，因此(X+1)mod 3=0，因此X+1可被3整除，又因为X肯定为奇数，所以X+1肯定能被2整除，因此X+1能被6整除。

18、 假设有 n 条鱼，它们从小到大依次编号为 1, 2, …, n 。我们规定，吃鱼必须要严格按顺序执行。也就是说，大鱼只能吃比自己小一级的鱼，不能越级吃更小的鱼；并且只有等到第 i 条鱼吃了第 i - 1 条鱼后，第 i + 1 条鱼才能吃第 i 条鱼。第 1 条鱼则啥都不能吃，只有被吃的份儿。我们假设，如果有小鱼吃的话，大鱼肯定不会放过；但是，保全性命的优先级显然更高，在吃小鱼之前，大鱼得先保证自己不 会被吃掉才行。假设每条鱼都是无限聪明的（并且它们也都知道这一点，并且它们也都知道它们知道这一点??），那么第 1 条鱼能存活下来吗？

答 案或许有些出人意料：当 n 是奇数时，第 1 条鱼将会存活下来；当 n 是偶数是，第 2 条鱼将会吃掉第 1 条鱼。为了证明这一点，让我们来考虑一些简单的情况。当 n = 1 时，第 1 条鱼显然活得自由自在；当 n = 2 时，第 2 条鱼将会吃掉第 1 条鱼，因为第 2 条鱼是无敌的，它不用担心自己会被吃掉。当 n = 3 时，第 2 条鱼不能吃第 1 条鱼，否则它将会被第 3 条鱼吃掉。有趣的事情发生在 n = 4 的时候，此时第 2 条鱼可以大胆地吃掉第 1 条鱼，因为根据前面的结论，它知道第 3 条鱼是不会吃它的??以此类推，当 n 是奇数时，这 n 条鱼将会和平相处；当 n 是偶数时，第 1 条鱼将会被第 2 条鱼吃掉，情况就化为了 n 为奇数时的稳定状态。

19、某人上午八点从山脚出发，沿山路步行上山，晚上八点到达山顶。不过，他并不是匀速前进的，有时慢，有时快，有时甚至会停下来。第二天，他早晨八点从 山顶出发，沿着原路下山，途中也是有时快有时慢，最终在晚上八点到达山脚。试着说明：此人一定在这两天的某个相同的时刻经过了山路上的同一个点。 这个题目也是经典中的经典了。把这个人两天的行程重叠到一天去，换句话说想像有一个人从山脚走到了山顶，同一天还有另一个人从山顶走到了山脚。这两个人一 定会在途中的某个地点相遇。这就说明了，这个人在两天的同一时刻都经过了这里。

20、甲从 A 地前往 B 地，乙从 B 地前往 A 地，两人同时出发，各自匀速地前进，每个人到达目的地后都立即以原速度返回。两人首次在距离 A 地 700 米处相遇，后来又在距离 B 地 400 米处相遇。求 A 、 B 两地间的距离。

答案： 1700 米。第一次相遇时，甲、乙共同走完一个 AB 的距离；第二次相遇时，甲、乙共同走完三个 AB 的距离。可见，从第一次相遇到第二次相遇的过程花了两个从出发到第一次相遇这么多的时间。既然第一次相遇时甲走了 700 米，说明后来甲又走了 1400 米，因此甲一共走了 2100 米。从中减去 400 米，正好就是 A 、 B 之间的距离了。

21、甲、乙、丙三人百米赛跑，每次都是甲胜乙 10 米，乙胜丙 10 米。假设三人全程匀速。则甲胜丙多少米？

答案是 19 米。“乙胜丙 10 米”的意思就是，等乙到了终点处时，丙只到了 90 米处。“甲胜乙 10 米”的意思就是，甲到了终点处时，乙只到了 90 米处，而此时丙应该还在 81 米处。所以甲胜了丙 19 米。

22、哥哥弟弟百米赛跑，哥哥赢了弟弟 1 米。第二次，哥哥在起跑线处退后 1 米与弟弟比赛，那么谁会获胜？

答案是，哥哥还是获胜了。哥哥跑 100 米需要的时间等于弟弟跑 99 米需要的时间。第二次，哥哥在 -1 米处起跑，弟弟在 0 米处起跑，两人将在第 99 米处追平。在剩下的 1 米里，哥哥超过了弟弟并获得胜利。

23、如果你上山的速度是 2 米每秒，下山的速度是 6 米每秒（假设上山和下山走的是同一条山路）。那么，你全程的平均速度是多少？

这是小学行程问题中最容易错的题之一，是小孩子们死活也搞不明白的问题。答案不是 4 米每秒，而是 3 米每秒。不妨假设全程是 S 米，那么上山的时间就是 S/2 ，下山的时间就是 S/6 ，往返的总路程为 2S ，往返的总时间为 S/2 + S/6 ，因而全程的平均速度为 2S / (S/2 + S/6) = 3 。

24、船在流水中逆水前进，途中一个救生圈不小心掉入水中，一小时后船员才发现并调头追赶。则追上救生圈所需的时间会大于一个小时，还是小于一个小时，还是等于一个小时？

这也是一个经典问题了。中学物理竞赛中曾出现过此题，《编程之美》上也有一个完全相同的问题。答案是等于一个小时。原因很简单：反正船和救生圈都被加上了 一个水流的速度，我们就可以直接抛开流水的影响不看了。换句话说，我们若以流水为参照系，一切就都如同没有流水了。我们直接可以想像船在静水当中丢掉了一 个救生圈并继续前行一个小时，回去捡救生圈当然也还需要一个小时。每当有人还是没想通时，我很愿意举这么一个例子。假如有一列匀速疾驰的火车，你在火车车 厢里，从车头往车尾方向步行。途中你掉了一个钱包，但继续往前走了一分钟后才发现。显然，你回去捡钱包需要的时间也是一分钟。但是，钱包不是正被火车载着 自动地往远方走吗？其实，既然你们都在火车上，自然就可以无视火车的速度了。前面的救生圈问题也是一样的道理。

25、某工厂每天早晨都派小车按时接总工程师上班。有一天，总工程师为了早些到工厂，比平日提前一小时出发步行去工厂。走了一段时间后，遇到来接他的小车 才上车继续前进。进入工厂大门后，他发现只比平时早到 10 分钟。总工程师在路上步行了多长时间才遇到来接他的汽车？设人和汽车都做匀速直线运动。

据说，这是一道初中物理竞赛题（初中物理有“运动”一章）。答案是 55 分钟。首先，让我们站在车的角度去想（正如前一题那样）。车从工厂出发，到半途中就遇上了总工程师并掉头往回走，结果只比原来早到 10 分钟。这说明，它比原来少走了 10 分钟的车程，这也就是从相遇点到总工程师家再到相遇点的路程。这就说明，从相遇点到总工程师家需要 5 分钟车程。现在，让我们把视角重新放回总工程师那里。让我们假设总工程师遇上了来接他的车并坐上去之后，并没有下令汽车立即掉头，而是让车像平日那样继续 开到他家再返回工厂，那么他到工厂的时间应该和原来一样。这说明，他提前出发的那一个小时完全浪费了。这一个小时浪费在哪儿了呢？浪费在了他步行到相遇点 的过程，以及乘车又回到家的过程。既然乘车又回到家需要 5 分钟，因此步行的时间就是 55 分钟了。

26、有一位隐居在深山老林的哲学家。一天，他忘记给家里唯一的时钟上发条了。由于他家里没有电话、电视、网络、收音机等任何能获知时间的设备，因此他彻 底不知道现在的时间是多少了。于是，他徒步来到了他朋友家里坐了一会儿，然后又徒步回到自己家中。此时，他便知道了应该怎样重新设定自己的时钟。他是怎么 做的？

很多人的第一想法或许是观察日出日落。在此，我们也假设通过太阳位置判断时间是不可靠的。 Update: 不少网友找到了此题的一个 bug 。在此我们假设，时钟是固定在墙上的，或者由于太重，无法直接带走。答案：别忘了，他家里的时钟并不是不走了，只是不准了而已。因此，他可以借助自己家里 的时钟，判断他此次出行一共花了多久。假设往返所花时间一样，再结合在朋友那儿看到的正确时间，他便能算出应该怎样调整自己的时钟了。

**六、数据库与系统知识**

1、mysql主从同步的实现原理？

2、对于web应用，当用户量增长之后一般怎么做？

3、myisam和innodb的区别，何时用哪种比较合适？

4、nosql相关系统的了解程度？dynamo，beandb，cassandra

**七、软素质**

1、假设你是一名学生，从图书馆借了10本书，2个月后准备去还书的时候发现书架上只有8本了，这个时候你怎么办？

2、假设你是吉利的老总，董事会希望做一款豪车打入美国市场，你准备怎么做？

3、假设你是一名货车司机，需要将一批货从北京运到武汉，途中遇到一个山洞，洞比车最高点矮了5cm，你怎么办？