**面试**

# 1 案例

## 1.1面试360

自我介绍

说一个你测过系统的功能 然后自己设计一下测试用例

后台批量导入订单（读取Excel文件—导入订单—生成订单号—提交至oms系统—审核）

导入文件的判断、数据格式的判断（为空、非法数据、重复数据、边界值）、文件格式、大小、订单数量限制、生成订单号规则、重复订单、重复提交同一文件、订单处理耗时、订单状态的判断、操作失败处理……

你发现的最有价值的bug

http协议请求响应都有什么内容 服务器端返回状态码 读取一个目录下所有后缀名为.txt的文件进行打印并统计 现场写

os模块都有什么函数功能

读取一个excel 一列是url 一列是状态码 写一个自动化进行检验

性能测试流程

你对windows系统的了解

window下安全卫士软件垃圾回收的原理

如果电脑内存使用过高 去哪找问题 分别说硬件软件

内存泄露内存溢出区别

## 1.2面试搜狗服务器端测试

个人介绍

字符串内容翻转输出尽可能多写方法

写一个接口自动化 写一个校验ipv4正确性的算法

功能测试用例设计

写一个登录页面测试用例设计

你帮我买一张火车票回家 根据这个做用例设计

浏览器收藏同步功能写详细功能测试用例

性能测试分析问题都有碰到哪些 具体怎么做分析 怎么样得到的这个结果，具体看的哪些指标得到的，为什么就判定它是问题

## 1.3高级测试开发服务端

自我介绍：主要在公司的成绩、职责

性能测试：

明确目标：tps、响应时间、吞吐量、并发用户数

制定测试计划：

测试环境部署：硬件、软件、网络、数据库基础数据

编写测试脚本：参数化、关联、检查点

测试策略：测试用例

执行脚本：

收集分析结果：

测试报告：

对被测试系统架构的了解：分库、分表、转发、负载均衡

linux：关于性能调优的命令、数据库影响到性能的东西、中间件与性能有关的

慢查询问题：慢查询日志、explain

性能测试调优，如何知道有性能问题？

cpu（进程-分布式计算）、IO（iostat、缓存）、内存（内存溢出）、数据库（慢查询、连接数）、缓存、算法、网络优化【资源紧缺】

什么是内存溢出，如何判断？

分配的内存无法回收，定期查看内存占用的变化，用lr做稳定性测试

线程/进程区别？

python伪线程GIL

线程调度、时间片轮换？

线程/进程间的同步，主线程怎么判断子线程是否执行结束？

isAlive() pid

堆、栈概念，区别、效率

操作系统的堆栈

数据结果的堆栈

链表与数组效率？为什么？

数组分配内存-空间连续的，大小固定==>导致增加、删除的速度比较慢

链表（单向链表、双向链表、循环链表），由指针指向下一个元素，空间不连续==>导致查询修改速度慢

selenium、appium 原理？

selenium驱动浏览器：

selenium1==>js驱动

selenium2==>webdriver的加入，直接调用浏览器api进行驱动

selenium3==>同selenium2

appium：client<==>server

bootstrap.jar

tcp、udp概念区别？

网络7层协议，每层的职责，http、tcp、udp、ip

http==>https 需要测试些什么内容？

vpn的原理？

安全测试，sql注入、xss攻击，安全工具有哪些（appscan、Wireshark）

持续集成，快速迭代

敏捷开发的理解？

linux相关：根据端口号找出其对应的服务部署到什么位置？

netstat、ps

与性能相关的命令：awk、sed

linux管道

数据库相关：索引

index与key的区别

如何查看数据库慢查询

接口相关：

如何做接口测试？

测试中要关注什么？

是否实现过接口测试框架、如何实现？

如何做支付方面测试

功能测试框架+性能+安全

算法：

## 1.4国美金融

1. 一套完整的测试生命周期？
2. 你所熟悉的测试用例设计方法有哪些？分别以具体实例说明在工作中的应用
3. 你认为做好测试用例设计工作的关键是什么？
4. Linux中一个文件的访问权限是755，其含义是什么？
5. Linux中如何从root用户切换到普通用户？
6. Linux中如果你正在运行一个进程，而且你觉得在退出账户时该进程不能结束，用什么命令，简单描述其语法？
7. 数据库：为管理业务培训信息建立3个表：

Student(S#,SN,SD,SA）分别代表学号、学员姓名、所属单位、学员年龄

Course(C#,CN)分别代表课程编号、课程名称

SC(S#,C#,G)分别代表学号、所选课程编号、学习成绩

1. 使用sql嵌套语句查询选修课程名称为‘数据结构’的学员学号和姓名？
2. 使用sql嵌套语句查询选修课程编号为‘C2’的学员姓名和所属单位？
3. 使用sql嵌套语句查询不选修课程编号为‘C5’的学员姓名和所属单位？
4. 查询选修了课程的学员人数
5. 查询选修课程超过5门的学员学号和所属单位
6. 简述软件自动化测试工具的原理？
7. 现有关系型数控库如下，请用SQL语句实现下列1-3小题：

Student(Sno,Sname,Ssex,Sdept)

Course(Cno,Cname,Credit)

SC(Sno,Cno,Grade)

题目一：检索”EN”专业学生所学课程信息，包括学号、姓名、课程名、分数

题目二：检索”Java”课程成绩高于90分的所有学生的学分数增加1.5

题目三：检索没学课程号为”C135”课程的学生信息，包括学号、姓名和专业

1. 简述负载测试、压力测试、性能测试之间的异同？
2. 谈谈软件测试技术，以及一个优秀软件测试人员应该具备的素质？

## 其他面试

# 1 请描述系统测试和用户验收测试区别？

## 系统测试：

系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。

## 用户验收测试：

它是技术测试的最后一个阶段，也称为[交付测试](http://baike.baidu.com/view/15018974.htm)。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

## 区别：

1. 验收测试是在系统测试执行完成之后进行的行为；
2. 系统测试目的是验证需求实现结果；验收测试是系统在生产环境能否按预期执行并完成用户需要的任务

# 2 测试准备活动包括那些？

测试过程中需要准备的有以下四大部分：

## 1. 测试文档准备

需求文档的准备并理解、测试工作量评估及测试计划制定、测试用例文档编写、测试报告等模板的准备。

## 2. 测试环境的搭建

系统需要运行环境的搭建，包括硬件、软件、网络等环境的准备和配置。

## 3. 测试数据

要根据实际系统运行环境造大量真实可能的数据。

## 4. 测试管理工具

包括缺陷管理工具或者项目管理工具等搭建

# 3 在测试结束后或者测试执行中是否会对测试结果进行分析，会分析哪些维度的内容及其目的？

## 测试结果分析内容：

分析BUG主要集中在哪些功能上，都是一些什么类型BUG

## 测试执行中目的：

通过分析的结果可以在集中出现BUG的功能上分配更多的时间进行测试，通过BUG的类型根据经验来定位问题并且判断系统的健壮性。

## 测试结束后目的：

对测试和开发人员进行考核。总结系统上常出现的和比较严重的BUG类型记录下来方便以后在项目中做BUG预防。

# 4 有哪些软件开发模型？并在不同开发模型下开展测试活动的差异进行描述。

## 瀑布模型：

传统的周期模型：计划、需求分析、概要设计、详细设计、编码以及单元测试、测试、运行维护等阶段。

## 原型化模型：

创建一个快速模型，在模型的基础上与用户直接进行讨论最终确定最后的模型及需求然后开发。

## 迭代模型：

把项目分成若干个阶段（初始、细化、构建、交付），每个阶段都是由若干个传统流程或者全部传统流程构成。

## 测试模型：

有V、W、H模型，在开发的模型当中，可以根据实际情况确定测试用什么模型，一般来说用H模型，测试是一个独立的流程，从需求执行就开始介入项目，可以最好的保证项目质量。

# 5 商业集团关于商店销售的的数据库有3个表：

商店（**商店号**，商店名，区域名，经理姓名）

商品（**商品号**，商品名，单价）

销售（**商店号**，**商品号**，**销售日期**，销售数量）

说明：

1. 加粗字段为主关键字
2. 涉及金额查询中：销售金额=销售.销售数量\*商品.商品单价
3. 查询不在“北京”、“上海”区域的商店信息
4. 查询商品单价在10到50之间，日销售数量高于20的商品名、单价、销售日期、销售数量，查询结果按单价降序。

一般sql分三个部分：

1. 要进行的操作（插入，更新，新建，删除，查询）
2. 要操作的表（多个表，单个表）
3. 要操作的范围（使用条件组合）

## 第一题

进行查询的操作查询的表是商品信息表，查询范围是区域名不包括北京和上海：

Select \* from 商店 where 区域名 not in (“北京”,”上海”);

## 第二题涉及到表关联：

表之间关联有

内关联（inner join）展示每个表中共有的数据

左关联（left join）展示左表中想要看的所有内容，右表中展示内容能匹配到就展示匹配到的，匹配不到的展示null

右关联（right join）展示右表中想要看的所有内容，左表中展示内容能匹配到就展示匹配到的，匹配不到的展示null

全关联（full join）左表右表所有内容都展示，匹配到的就展示匹配到的，匹配不到的就展示

Null

从题中看出查询结果：商品名、单价、销售日期、销售数量（按单价降序）；

查询涉及到的表：是商品表和销售表

查询条件是：商品单价在10到50之间；日销售数量高于20

从题意看出要使用内关联。

Select 商品.商品名,商品.单价,销售.销售日期,sum(销售.销售数量) from 商品 inner join 销售 on 商品.商品号=销售.商品号 group by 销售.商品号,销售.销售日期 having 商品.单价>10 and 商品.单价<50 order by 商品.单价 desc;

使用sql拼接的方式进行查询：

with dt as(select 商品号,单价  from 商品 a where 单价 >=10 and 单价<=50 )  
select a.单价,b.商品名称,b.销售数据量, 销售日期（日） from dt a in2ner join(  
select 商品号,商品名称,Sum(销售数量) as 销售数据量，CONVERT(varchar(100), 销售日期, 23) as 销售日期（日）  from 销售 group by 商品号,CONVERT(varchar(100), 销售日期, 23),商品名称having Sum(销售数量)>20 ) b on a.商品号 = b.商品号group by a.单价,b.商品名称,b.销售数据量，销售日期（日）order by a.单价 desc

# 6 某网站注册用户名规则：

1. 长度6-15
2. 与字母数字下划线中至少两种组合
3. 首字母必须大写字母或者数字

对该功能进行测试，罗列测试用例最小合集

这道题核心思想就是考察等价类和边界值：

## 首先分析每条需求：

1长度测试使用边界值和等价类：有效：6，10，15 无效：5，16，0，999

2组合使用等价类：无效的：纯数字，纯字母，纯下划线；非法字符 有效的：数字字母下划线；数字字母；数字下划线；下划线字母

3首字母使用等价类：有效：大写字母；数字 无效：小写字母；下划线；非法字符

## 综合1、2、3的分析结合n-wise思想：

1有效2有效3有效

1无效2无效3无效

1有效2无效3有效

1有效3无效2有效

2有效1无效3有效

2有效3无效1无效

3有效1无效2无效

3有效2无效1有效

## 最后：

测试数据准备就以每三种情况中最多情况为准，全部数据覆盖到就OK。

# 7 您在工作中兼容性测试涉及到哪些硬件/软件？进行举例，描述该场景下兼容性测试的要点

## 硬件：

显示器；手机型号

## 软件：

操作系统；浏览器；防火墙；系统补丁

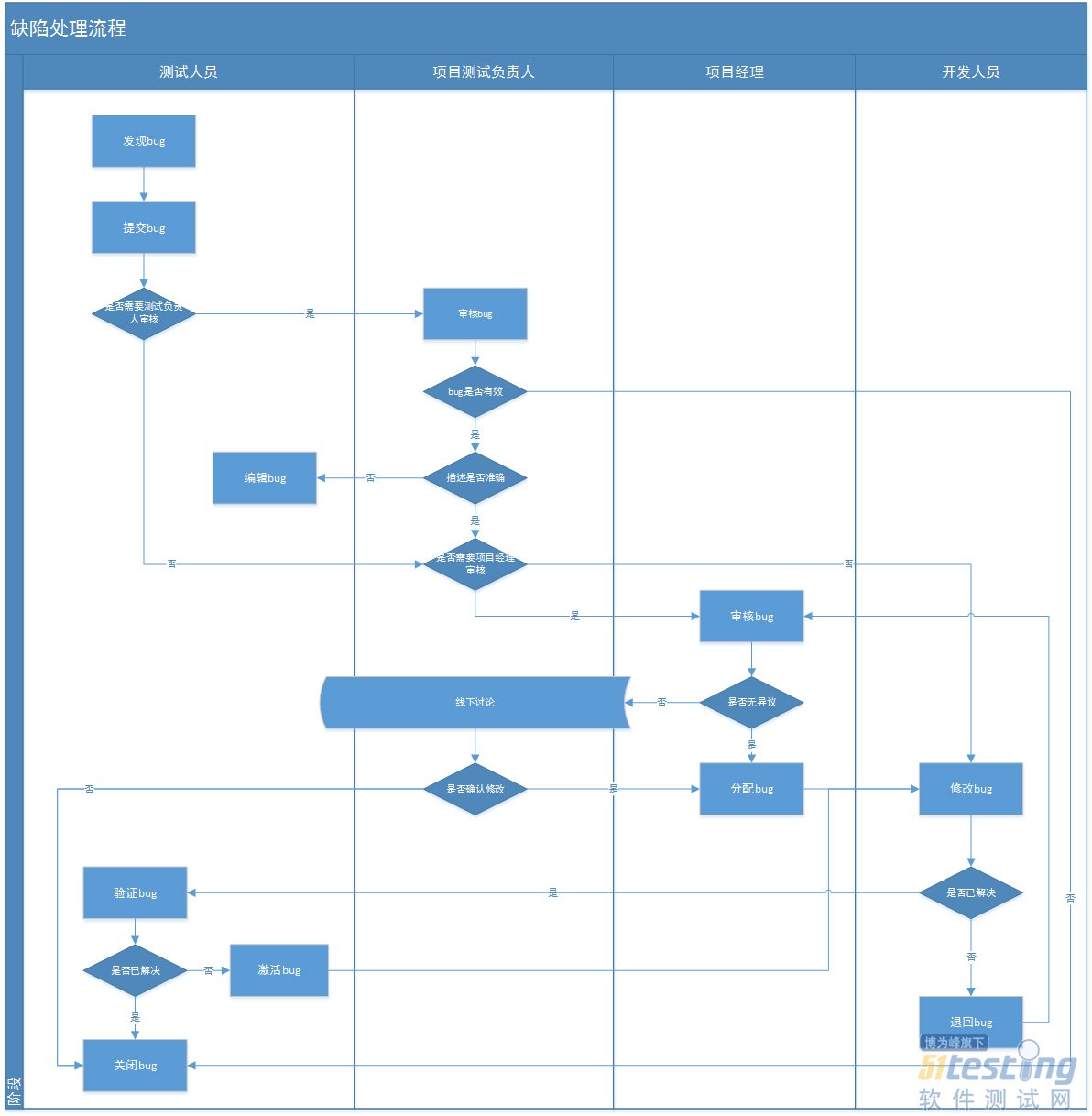
## 移动端测试要点：

不同浏览器种类及版本号/不同手机型号/下用户注册登录退出、系统主流程等功能的执行及展示；有APP的再考虑安装卸载和升级降级覆盖

## PC端测试要点：

不同显示器分辨率/不同浏览器种类及版本/不同防火墙/不同操作系统及版本号/系统补丁下用户注册登录退出、系统主流程等功能的执行及展示；有客户端考虑安装卸载和升级降级覆盖。

# 8 某缺陷管理工具状态流转图，对该工作进行测试，能设计多少测试用例，描述设计思路



## 涉及到具体路径的都用白盒测试的思想进行测试：

白盒测试覆盖方法有：[语句覆盖](http://baike.baidu.com/view/1741053.htm)、[判定覆盖](http://baike.baidu.com/view/953374.htm)、[条件覆盖](http://baike.baidu.com/view/1289596.htm)、判定/条件覆盖、[条件组合覆盖](http://baike.baidu.com/view/229032.htm)和[路径覆盖](http://baike.baidu.com/view/2193850.htm)

# 9 某移动端炒股APP核心功能包括不限于：

1. 用户管理（用户登录注册、用户信息管理、开户榜卡、充值、取现）
2. 行情资讯（与第三方行情服务器通讯获得并显示行情信息）
3. 交易（与交易所系统信息交互进行股票买卖）

对该系统进行测试，您会做哪些类型的测试？

## 这类题一般就是往测试框架里面套：

比如APP就需要考虑（小括号里都是再基于PC端基础上移动端额外需要注意的情况）：

需求文档测试

界面测试

功能测试

算法测试

可靠性（弱网络，内存不足，强制关闭，接打电话，后台运行）

易用性（手机快捷键，小键盘）

可维护性

安全测试

竞品测试

前端性能（耗流量；耗电量；占内存）

后端性能（压力测试，稳定性）

兼容性（不同手机型号下卸载安装升级降级覆盖）

# 10面试总体分类

## 测试设计

功能测试框架

Bug预防

探索性测试

## 线程

单线程

多线程

**每个线程可以结合它所属进程的优先级，并使用System.Threading.Thread类中类型为ThreadPriority的Priority{get;set;}属性定义各自的优先级。**

进程

**可以使用Process类中的类型为ProcessPriorityClass的PriorityClass{get;set;}属性为进程赋予一个优先级。**

线程池

**线程池是一种多线程处理形式，处理过程中将任务添加到队列，然后在创建线程后自动启动这些任务。线程池线程都是后台线程。每个线程都使用默认的**[**堆栈**](http://baike.baidu.com/item/%E5%A0%86%E6%A0%88)**大小，以默认的优先级运行，并处于多线程单元中。如果某个线程在**[**托管代码**](http://baike.baidu.com/item/%E6%89%98%E7%AE%A1%E4%BB%A3%E7%A0%81)**中空闲（如正在等待某个事件）,则线程池将插入另一个**[**辅助线程**](http://baike.baidu.com/item/%E8%BE%85%E5%8A%A9%E7%BA%BF%E7%A8%8B)**来使所有处理器保持繁忙。如果所有线程池线程都始终保持繁忙，但队列中包含挂起的工作，则线程池将在一段时间后创建另一个辅助线程但线程的数目永远不会超过最大值。超过最大值的线程可以排队，但他们要等到其他线程完成后才启动。**

单线程多线程进程区别

**进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。**

## 接口

接口框架的搭建

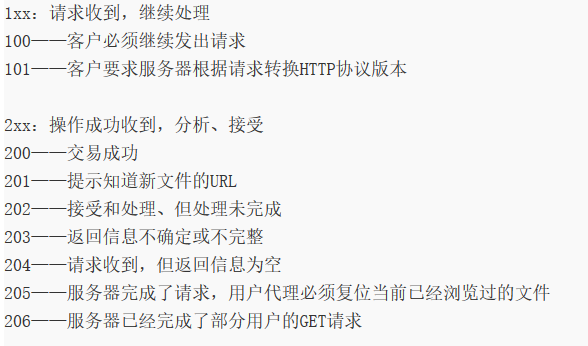
http协议

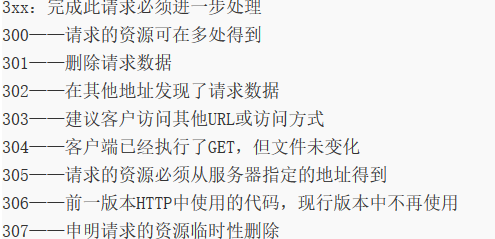
请求和响应内容

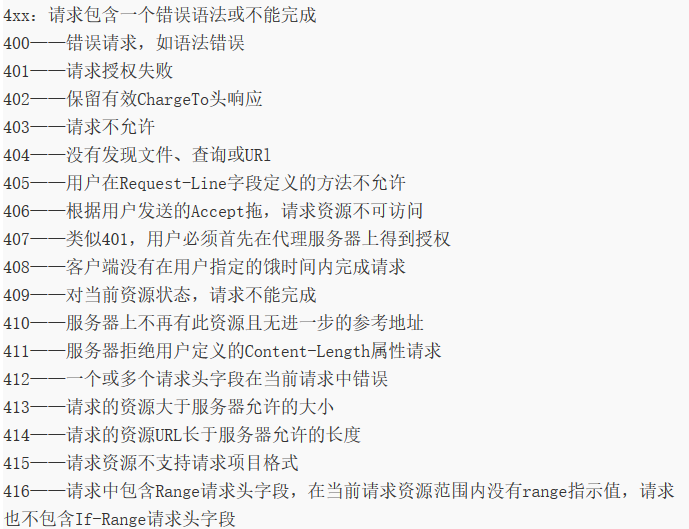
请求头：accept：encoding，language，connection，cookie，host，refer，user-agent

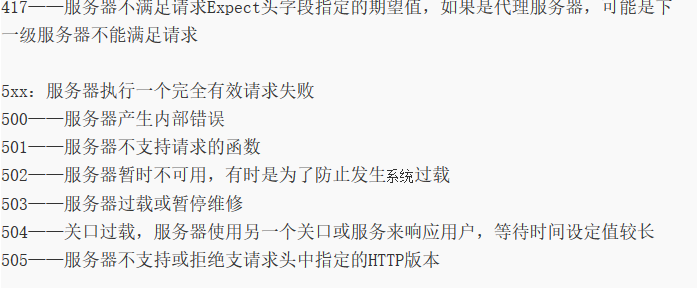
响应头：Content-Encoding，connection，date

服务器端状态码









Get方法和post方法不同

**POST把提交的数据则放置在是HTTP包的包体中。**

**GET请求的数据会附在URL之后（就是把数据放置在HTTP协议头中），以?分割URL和传输数据，参数之间以&相连**

**GET方式提交的数据最多只能是1024字节，理论上POST没有限制Cokkie和session区别**

POST是没有大小限制的，HTTP协议规范也没有进行大小限制

URL不存在参数上限的问题，HTTP协议规范没有对URL长度进行限制

Get是向服务器发索取数据的一种请求，而Post是向服务器提交数据的一种请求

浏览器内核

**浏览器内核又可以分成两部分：渲染引擎(layout engineer 或者 Rendering Engine)和 JS 引擎。它负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入 CSS 等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。JS 引擎则是解析 Javascript 语言，执行 javascript 语言来实现网页的动态效果。**

Gecko ，webkit，Blink，

## APP

前端

后端

前后端交互（同步异步）

## 建设性意见

看门狗构建

挖掘过载问题，从上提出优化

保障产品可靠性提出双机/集群方案，提出备份容灾备份方案

## 自动化测试能力

1.实际自动测试经验，是否独立搭建自动化框架

2.如何保证用例之间的去耦合性

3.UI自动化，如何解决工作量大的问题

4.是否做过自动化生成测试用例

5.持续集成相关工作，哪些做的好，哪些不好

## 代码能力

算法

数据结构

### 内存溢出和内存泄漏区别

内存溢出 out of memory，是指程序在申请内存时，没有足够的内存空间供其使用，出现out of memory；比如申请了一个integer,但给它存了long才能存下的数，那就是内存溢出。

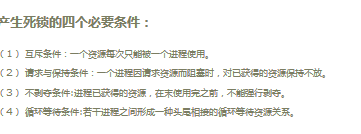
内存泄露 memory leak，是指程序在申请内存后，无法释放已申请的内存空间，一次内存泄露危害可以忽略，但内存泄露堆积后果很严重，无论多少内存,迟早会被占光。

memory leak会最终会导致out of memory！

内存溢出就是你要求分配的内存超出了系统能给你的，系统不能满足需求，于是产生溢出。   
    内存泄漏是指你向系统申请分配内存进行使用(new)，可是使用完了以后却不归还(delete)，结果你申请到的那块内存你自己也不能再访问（也许你把它的地址给弄丢了），而系统也不能再次将它分配给需要的程序。一个盘子用尽各种方法只能装4个果子，你装了5个，结果掉倒地上不能吃了。这就是溢出！比方说栈，栈满时再做进栈必定产生空间溢出，叫上溢，栈空时再做退栈也产生空间溢出，称为下溢。就是分配的内存不足以放下数据项序列,称为内存溢出.

### 死锁：

死锁是指在一组进程中的各个进程均占有不会释放的资源，但因互相申请被其他进程所站用不会释放的资源而处于的一种永久等待状态。



## 算法

### 查找算法

### 2. 二分查找

#### 说明：元素必须是有序的，如果是无序的则要先进行排序操作。

基本思想：也称为是折半查找，属于有序查找[算法](http://lib.csdn.net/base/datastructure)。用给定值k先与中间结点的关键字比较，中间结点把线形表分成两个子表，若相等则查找成功；若不相等，再根据k与该中间结点关键字的比较结果确定下一步查找哪个子表，这样递归进行，直到查找到或查找结束发现表中没有这样的结点。

### 3. 插值查找

#### 在介绍插值查找之前，首先考虑一个新问题，为什么上述算法一定要是折半，而不是折四分之一或者折更多呢？

#### 打个比方，在英文字典里面查“apple”，你下意识翻开字典是翻前面的书页还是后面的书页呢？如果再让你查“zoo”，你又怎么查？很显然，这里你绝对不会是从中间开始查起，而是有一定目的的往前或往后翻。

#### 基本思想： 基于二分查找算法，将查找点的选择改进为自适应选择，可以提高查找效率。当然，差值查找也属于有序查找。

### 4. 斐波那契查找

#### 基本思想： 也是二分查找的一种提升算法，通过运用黄金比例的概念在数列中选择查找点进行查找，提高查找效率。同样地，斐波那契查找也属于一种有序查找算 法。

#### 5.1 最简单的树表查找算法——二叉树查找算法。

基本思想： 二叉查找树是先对待查找的数据进行生成树，确保树的左分支的值小于右分支的值，然后在就行和每个节点的父节点比较大小，查找最适合的范围。 这个算法的查找效率很高，但是如果使用这种查找方法要首先创建树。

### 6. 分块查找

#### 分块查找又称索引顺序查找，它是顺序查找的一种改进方法。

#### 算法思想： 将n个数据元素"按块有序"划分为m块（m ≤ n）。每一块中的结点不必有序，但块与块之间必须"按块有序"；即第1块中任一元素的关键字都必须小于第2块中任一元素的关键字；而第2块中任一元素又都必须小于第3块中的任一元素，……

#### 算法流程：

#### step1 先选取各块中的最大关键字构成一个索引表；

#### step2 查找分两个部分：先对索引表进行二分查找或顺序查找，以确定待查记录在哪一块中；然后，在已确定的块中用顺序法进行查找。

### 哈希查找

#### 算法流程：

#### 1）用给定的哈希函数构造哈希表；

#### 2）根据选择的冲突处理方法解决地址冲突；

#### 常见的解决冲突的方法：拉链法和线性探测法。 详细的介绍可以参见： [浅谈算法和数据结构: 十一 哈希表](http://www.cnblogs.com/yangecnu/p/Introduce-Hashtable.html) 。

#### 3）在哈希表的基础上执行哈希查找。

#### 哈希表是一个在时间和空间上做出权衡的经典例子。如果没有内存限制，那么可以直接将键作为数组的索引。那么所有的查找时间复杂度为O(1)；如果没有时间限制，那么我们可以使用无序数组并进行顺序查找，这样只需要很少的内存。哈希表使用了适度的时间和空间来在这两个极端之间找到了平衡。只需要调整哈希函数算法即可在时间和空间上做出取舍。

### 6. 分块查找

分块查找又称索引顺序查找，它是顺序查找的一种改进方法。

算法思想： 将n个数据元素"按块有序"划分为m块（m ≤ n）。每一块中的结点不必有序，但块与块之间必须"按块有序"；即第1块中任一元素的关键字都必须小于第2块中任一元素的关键字；而第2块中任一元素又都必须小于第3块中的任一元素，……

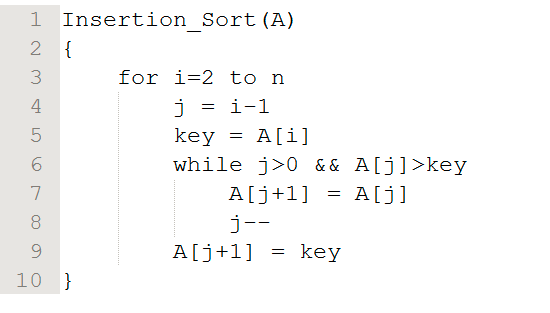
算法流程：

step1 先选取各块中的最大关键字构成一个索引表；

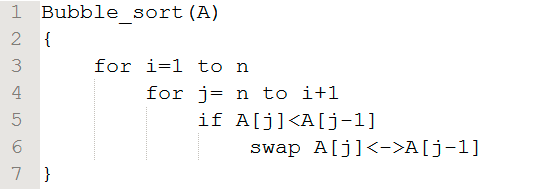
step2 查找分两个部分：先对索引表进行二分查找或顺序查找，以确定待查记录在哪一块中；然后，在已确定的块中用顺序法进行查找。

## 排序算法

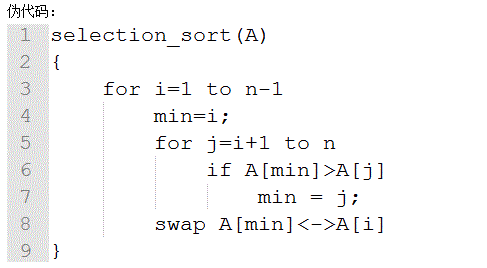
### 一、插入排序



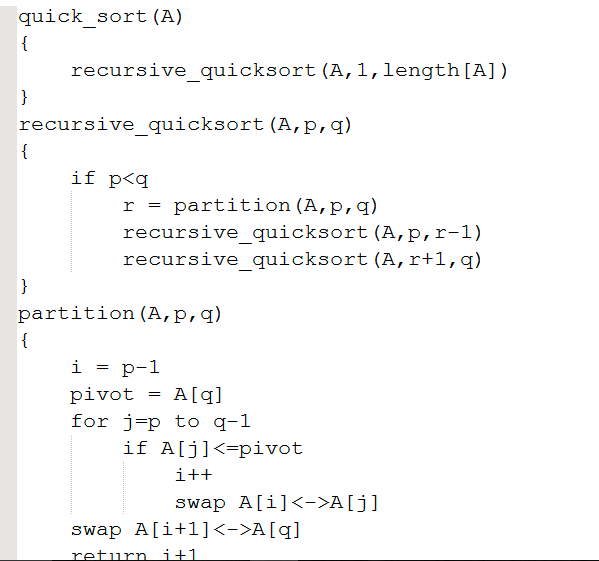
### 二、冒泡排序

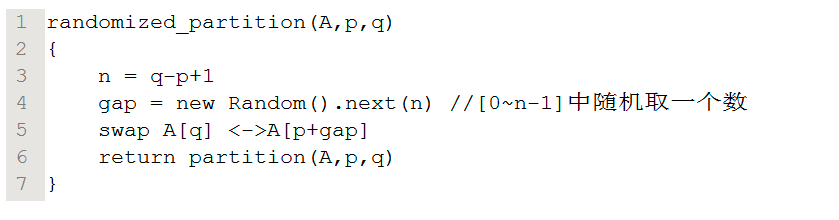


### 三、选择排序



### 快速排序





### 桶排序

### 堆排序

优先队列应用于进程间调度、任务调度等。

### 基数排序

### 计数排序

## 实战

项目做持续集成，写一个脚本完成linux下测试环境的自动化搭建

安装包规格

## 机试

最大公约数

最小公倍数

## 数据库

DDL DML对应sql

查询，聚合，存储过程

## 网络

IP

网关

路由

根据子网掩码算网关

NAT

VPN

交换机配置

## 项目流程

敏捷流程

持续集成

代码review

结对编程

测试人员在敏捷项目怎么做

## 性能测试

性能测试流程

性能测试定位发现的问题

Lr的使用（lr脚本组成）

## 典型例题

Web页面出现空白页怎么定位问题

Web页面响应过慢怎么定位问题

从一个浏览器输入url到形成页面经历了什么

你所测试的系统架构

**以下代码输出结果？**

def extendList(val, list=[]):

list.append(val)

return list

#调用extendList函数，这个函数调用中并没有list形参，所以list使用默认值[]，此时调用完extendList函数后，list=[10]

list1 = extendList(10)

#调用extendList函数，list=[]，但是此时的数组list和list1中的数组list已经不是同一个list了，可以通过id(list)得知，调用完extendList函数后list=[123]

list2 = extendList(123,[])

#调用extendList函数，这个函数调用中并没有list形参，所以此时list的值是和函数extendList中的list是一致的，由于list1=extendList(10)所以list=[10],所以再次调用完extendList函数后，#list=[10,'a']

list3 = extendList('a')

print "list1 = %s" % list1,id(list1) #[10,'a']

print "list2 = %s" % list2,id(list2) #[123]

print "list3 = %s" % list3,id(list3) #[10,'a']

def div1(x,y):

print("%s/%s = %s" % (x, y, x/y))

def div2(x,y):

print("%s//%s = %s" % (x, y, x//y))

div1(5,2)

div1(5.,2)

div2(5,2)

div2(5.,2.)

#coding=utf-8

class Parent(object):

x = 1

class Child1(Parent):

pass

class Child2(Parent):

pass

print Parent.x, Child1.x, Child2.x #1 1 1

Child1.x = 2 #给Child1这个类添加一个类变量并且赋值为1

print Parent.x, Child1.x, Child2.x #1 2 1

Parent.x = 3 #修改Parent的类变量x=3

print Parent.x, Child1.x, Child2.x #3 2 3

去哪儿测试笔试题

Liunx：

一、基础

1、 linux下查看系统任务状态的命令是？ ps -ef

2、 Linux下查看系统进程中是否有tomcat在运行的命令是？ps –ef|grep “java”

3、 Linux下查看系统硬盘空间的命令是？ df -h

4、 Linux下强制删除有多级子目录（含文件）的目录/home/share的命令是？

rm –rf /home/share

二、Shell

1、 将当前目录中所有后缀为.log的日志行数统计追加到当前目录的codelines.txt文件中并复制到/server/cache/目录下

cat \*.Log |wc -l >> condelines.txt

cp ./condelines.txt /server/cache/

网络：

1、 http返回值的含义：

"304" : 自从上次请求后，请求的网页未修改过。服务器返回此响应时，不会返回网页内容。

"404" : 页面找不到

"500" : 服务器内部错误

"502" : 错误网关

gzip是应用在http header的哪个参数中的

Content-Encoding gzip

2、 网站测试中cache技术的测试点都有哪些？

缓存大小／过期和更新策略效率/读取效率／并发性能／缓存数据更新正确性／缓存稳定性/缓存初始化的效率

性能：

一、LoadRunner：

1、 等待1分钟时间用什么函数 lr\_think\_time(60);

2、将内容输出用什么函数 lr\_log\_message

二、服务器和客户端的性能分析都从哪些角度来进行

服务器端主要是验证高并发的时候cpu 内存 io 的情况

数据库增删改查是否达到性能要求，以及缓存的命中率和效果，索引的处理能力，慢查询情况。

客户端主要看cpu占用／内存／网络连接/网络资源占用／本地io／本地日志输出策略

DB：

一、表Book结构及前19条数据如下：

sid name type ver desc

1 中国科技之光 科技 1 你试试？&nbsp;

2 中国科技之光 科技 2 你试试？&lt;

3 中国科技之光 科技 3 你试试？&gt;

4 中国科技之光 科技 4 你试试？&quot;

5 中国科技之光 科技 5 你试试？&nbsp;我会的！

6 中国艺术之光 艺术 1 你试&nbsp;哈飞"？

7 中国人文之光 人文 1 你试试？ 我会的！

8 中国体育之光 体育 1 你试"哈飞"？

9 中国生物之光 生物 1 你试试&lt;去哪儿>?

10 中国科技之光 科技 6 你试试&lt;去哪儿&gt;？

11 中国科技之光 科技 7 你试试&lt;去哪儿>?

12 中国艺术之光 艺术 2 你试试<去哪儿>？

13 中国人文之光 人文 2 你试试<去哪儿&gt;?

14 中国体育之光 体育 2 你试试？

15 中国体育之光 体育 3 你试试？我试试！

16 中国生物之光 生物 2 你试试？

17 中国艺术之光 艺术 3 你试试？不会的！

18 中国人文之光 人文 3 太多了&nbsp

19 中国科技之光 科技 8 搜索@lt出来

… … … … …

1、 请用1个sql获取图书type类型都有哪些（要求不重复）

select distinct type from book

2、 请用1个sql获取每类图书type都有多少本

select count（\*）from book group by type

3、 请用1个sql获取超过3本的图书类型及对应数量

select count（\*）as ct from book group by type having ct>=3

4、 请用1个sql语句获取同一类图书类型type中出版版本最新的一组数据

select \* from ct where ver in (select max(ver) from ct group by type)

5、 请用1个sql语句获取描述desc中包含html转义字符的记录【例如：&nbsp;&quot;&gt;&lt;】

SELECT \* FROM ct WHERE desc REGEXP '&'

6、 请用1个sql语句获取描述desc中包含三个英文字母连写记录

SELECT \* FROM ct WHERE desc regexp "[a-z][a-z][a-z]"

7、 如果图书数据很多，执行上述查询速度不是很理想，请说出你对数据库表的优化建议

这个表的最大问题在于desc字段的区分度不够，一方面在表的设计上尽量这个字段内容要不一致尽量提高区分度，另一方面可以通过在desc字段简历索引优化查询，建立的索引需要使用字段部分索引方式，如desc字段前三个词经常重复

例如

Alter table ct add index idx\_desc(desc[3:])

表示从第三个字符开始创建索引

二、请写出truncate和Delete、Drop表数据的差别？

truncate 会马上释放占用空,清除表结构

delete属于DML,删除表内数据，不删除表结构

drop、truncate都是DDL语句(数据定义语言),执行后会自动提交。原数据不放到rollback segment中,不能回滚. 操作不触发trigger

drop 语句将删除表的结构被依赖的约束(constrain)、触发器(trigger)、索引(index)；依赖于该表的存储过程/函数将保留,但是变为 invalid 状态。

一、基础

1、请写出下面代码的输出

private static void addWorld(String msg) {

msg = msg + "world";

}

public static void main(String[] args) {

String msg = "Hello ";

addWorld(msg);

System.out.println(msg);

}

输出：“Hello”

2、 请写出String类的包名

import java.lang.String;

3、请写出下面代码的输出

try {

System.out.println("try");

return;

} finally {

System.out.println("finally");

}

输出“try \n finally”

4、 请优化下面代码的性能

String line = “”;

for (int i = 0; i < 10000; i++) {

line += “item\_” + i;

}

优化：

Stringbuffer line＝“”；

for (int i = 0; i < 10000; i++) {

line .append(i);

}

5、IdentityHashMap m = new IdentityHashMap();

m.put("abc", "a");

m.put("abcd".substring(0,3), "a");

m.put(new String("abc"), "a");

m.put(null, "a");

请问 m.size() 是 ?

输出：4

二、程序

1. 写程序实现递归删除一个目录下的文件和文件夹

#coding=utf-8

import os

def removeDir(dirPath):

if not os.path.isdir(dirPath):

return

files = os.listdir(dirPath)

try:

for file in files:

filePath = os.path.join(dirPath, file)

if os.path.isfile(filePath):

os.remove(filePath)

elif os.path.isdir(filePath):

removeDir(filePath)

os.rmdir(dirPath)

except Exception, e:

print e

2. 请使用一种算法为下面的数组由小到大排序

int[] items = new int[]{1,3,5,7,9,2,4,6,8,0};

# -\*- coding: utf-8 -\*- #

numbers=[1,3,5,7,9,2,4,6,8,0]

for i in range(len(numbers)):

for j in range(i,len(numbers)):

if numbers[i]>numbers[j]:

numbers[i],numbers[j]=numbers[j],numbers[i]

for number in numbers:

print number

测试案例

一、如何测试只带1个用户名、密码框的登录界面？

1、请列举测试点及方法

2、 如果后台是使用msql2000数据库，验证登陆密码的程序是这样写的，请设计test case可以随意输入密码登陆

login页面代码

……

<form action="verify.asp" method="post" name="login">

用户名<input type=text name=name value="" maxlength="20">

密码<input type=password name=pwd value="" maxlength="20">

<input type=submit name=bt value="确认">

<input type=reset name=bt value="重置">

</form>

……

verify页面代码

dim rs,sql

dim name,pwd

name=request.form("name")

pwd=request.form("pwd")

if name="" or pwd="" then

response.redirect "login.asp"

end if

……

'关于身份验证的代码

sql="select \* from user where name='"&name&"' and pwd='"&pwd&"'"

……

二、产生用户界面乱码的原因都有哪些？

网页编码有问题

数据转码有问题

数据库存储有问题

数据本身就是乱码

印象深刻的bug：

1. 在做container与host环境融合时，部分实现的界面出现同一个activity被重复创建的问题，结果导致操作越多，界面会越来越k卡顿。

简单介绍自己

工作经历

最近工作中出现的问题及解决方式

维护测试用例（自动化脚本）

学习积累，如何运用到实际的工作中

自己作为测试工程师的优势在哪里

如何说明自己的学习能力、理解需求的能力、处理问题能力

安排测试计划

1. 问：“假设你所在的团队负责研发一款手机计算器程序（或者一直笔，一部电梯，一块表，一台银行ATM机），你是这款产品的测试负责人，你准备怎么开展工作？ ”

如果应聘者没有系统了解科学的项目测试理论，就很容易因以前的工作模式陷入思维定势，无法自拔。这类问题的流程：1.明确测试任务 2.分析测试范围 3.制定测试计划和测试用例

在回答该问题之前，先应该向面试官发问，明确产品支持的手机平台，之后，才能有的放矢的开展具体的设计（或者即使不问面试官支持哪些平台，在回答的时候也要说清楚先跟团队确定运行的平台）。再比如，应该明确产品的研发周期等信息，只有了解了项目进度安排等信息，才能制定有效的测试策略，在测试的深度和项目开发时间要求上取得较好的平衡。比如，有的项目是时间驱动的(Date-Driven)，这类项目的特点是预先制定发布时间，要求到了那天，产品就一定要发布，对这类项目，我们在设计测试计划时，就应该更多的考虑解决和项目发布相关的质量问题；另外有些项目，可能是质量驱动的(Quality-Driven)，这类项目的特点是对发布时间没有强行的规定，但要求产品的质量必须达到一定的指标，并且需要在发布以后，实时监控产品质量，那么，在测试中，我们不仅要做好项目当下版本的测试工作，还需要考虑构建长期、高效地测试系统和平台，保障产品质量能够实时度量。另外，明确产品的功能设计、产品的核心竞争力、可用的测试资源等信息，对于接下来做产品测试都是至关重要的。

开展测试前需要明确的问题：测试任务的范围是什么？开发为什么要做这些改动？这些改动是开发自己提出来的还是客户要求的？如果客户要求的客户的关注点在哪里？这次改动具体改了什么内容？怎么改的？你觉得这样的改动合理吗？改动以前是什么样子的？改动原因、改动内容和改动方法？

2. 问题：假设你是QQ这个产品的测试负责人，你怎么去测试QQ传文件这个功能？说一下测试点，你可以发挥自己的想象力，不必局限于它现有的功能。

两种思路：一种是按照传文件的流程（客户端A-网络-服务器-网络-客户端B），一种是是按照测试框架回答（比如系统的说明从UI、功能、性能、兼容性、安装部署、服务器端、网络、安全。

3. 与开发沟通确认问题

阅读开发修改日志，阅读代码改动。1 为什么要改动 2 修改前后的逻辑变化是什么 3 修改的细节 4 改动后的影响 5 确定回归测试范围

4. 问： 测试人员提交了一个bug，但开发人员不认同你，你改怎么办？

这个问题考察面比较固定，比如能反映一个人沟通方式（是否懂得正确的推进方式），工作主动性（比如沟通前是否先对一下需求和设计），对待bug是否有自己的立场和坚持等。

5. 问：为什么离职？

考察应聘者对待加班、薪资、与同事、领导相处关系等。 避免招进喜欢搬弄是非的人（说是非者必是是非之人）。

6. 问：说一下软件测试流程

这个问题应聘不同等级的测试需要不同的回答。若是来应聘高级测试，我希望听到的回答是在各个阶段有自己的独到理解，比如在需求分析阶段的分析，在bug预防上，在事后总结上。

1 为什么会选择做测试这份工作？

（重点考察测试人员对待测试工作的态度及是否有发展潜力）

2 你最近3-5年的职业规划是什么？

重点考察测试人员的职业发展方向是否与当前职位招聘相符? 从其中可以侧面看出来其员工稳定性。

3 请说出一个你以前参与项目，对你测试经验提升很高的，具体是哪方面？

重点考察测试人员在以往的测试工作中能力提升方面，有哪些？然后重点询问此部分内容，是否测试经验增长，具备一定的深度？

4 公司产品，具体应用什么编程技术？具体的架构是？具体的应用场景有哪些？

重点考察测试人员对以往的工作所负责的产品测试，是否具备一定的深度！通常我都是让面试者自己讲述或是在纸上画出具体系统架构的图！

5 公司测试团队的规模如何，具体你所处的角色是什么？

重点考察测试人员在以往的公司测试团队中，具体的工作职责，评判其工作是否与当要求职位是否符合？是否有哪些优缺点？

6 一个项目测试结束，有没什么经验总结？如果有，具体是如何开展的？

重点考察测试人员对自己能力提升方面，有没有提高总结的地方，从项目中吸取的经验与教训。从中可以看出来，测试人员是否属行自我驱动型人才！

7 工作过程中，是否曾主动总结过什么东西与他人分享？具体案例？

8 你认为做好软件测试工作，哪个环节的工作做好最重要？

需求理解

9 通常做测试时会碰到，提交的某个bug开发人员不认同你的观点？这时你如何办？

重点考察测试人员是否坚持自已的价值观？是否具备协调沟通处理问题能力？工作主动性（比如沟通前是否先对一下需求和设计）？

10 在多年的测试过程中是否有遇到偶发的Bug，是如何回归偶发Bug的？

自动化实现复现bug

尽可能的寻找必现步骤、收集log

更换版本，手机，操作步骤

11 从事测试多来，是否有感到自己哪些方面比较欠缺的？

12 有没有看过什么测试书，具体是哪本？带给你的收获是？

重点考察测试人员是否为测试这个职业肯付出多少？从中也可以看出这个测试人员是否上进心？是否有求知心？我的定义是如果哪个应聘者来面试时，都没系统的看过一本测试书籍，基本上不会录取！

13 如果安排一项测试技术研究工作，你如何应对？

重点考察测试人员是否具体测试技术专研精神？是否喜欢接受挑战？是否属于以后培养骨干对象？

14 某个项目上线后，出现问题，恰巧你是负责的，你如何应对这突如其来的事件？

重点考察测试人员应对问题的压力，责任感，及如何处理项目上线后的技术问题及应对解决能力。

15 周末放假有什么业余爱好？

重点考察面试测试人员性格特质，测试工作本身就是复杂且富有技术性的工作，而且不同的职位所需要的测试人员性格品质差异性很大。

16 特定测试技术考察：性能测试，安全性测试，自动化测试等以前有开展过没？如果有，具体是如何实施的？

重点考察测试人员技术能力，是否在各方面都有所涉及？或是在各方面技术上都有一定深度？当然从中也能看出一个测试人员是否属于是技术路线发展方向！

17 文档编写能力如何？

18 你自己所期待加入的测试团队是什么样的？

重点考察测试人员在以前测试团队中有哪些不协调？当然最重要的是也能提供给你一些信息，这个员工以后如何更好的管理与沟通！

19 告诉我们你是怎么接手一个新测试项目的吧。你首先会做什么事，问哪些问题？

性能测试

1. 性能测试的意义和作用，说出因为性能不良造成的质量事故？

2. 如何进行性能测试，整体的性能测试流程是什么？ 确定需求-制定计划和策略-准备环境（干净的，数据）-编写脚本-设计测试场景-运行-监控执行-分析测试结果

3. 性能测试的难点在哪里？如何克服？

4. 如何选择性能测试工具？

5. 如何确定性能测试团队的人力资源需求？

6. 性能测试会用到哪些知识？

7. 系统瓶颈如何分析，列出系统瓶颈？ ---最笨的、最基本的方法是按照一定的规则压并发，看日志。

8. 如何分析一个linux系统存在了内存不足？

9. 如何证明一个linux系统中的程序存在内存泄露？

10. 如何证明一个linux系统中的IO能力存在瓶颈？

11. loadrunner中的unique会使用在哪种性能测试场景需求中使用？

12. 请说出loadrunner中的关联是怎么回事？

13. 如果把性能测试工程师分为三级，你觉得如何划分？

14. 性能测试和压力测试是什么关系？ 压力测试和负载测试是什么关系？

15. loadrunner脚本出现乱码怎么解决？录制脚本时无法打开浏览器如何解决？如果脚本在运行中报错，如何分析原因和解决？

16. 录制脚本的2个模式分别是什么？他们的区别

17. loadrunner的随机化用在什么场景？

18. 性能测试的实现原理？

19. 如何制定一个性能测试的指标？哪些指标是核心的？

20. 线程和进程有什么区别？调用接口是线程还是进程（线程）？

21. loadrunner的并发数和在线用户数是个什么关系？

22. 说出你见过的一些设计优秀的网站系统框架，并且说明他们的缺点和优点？

23. 如何找到大型系统中的最大瓶颈点？

24. 你如果怀疑某段程序有问题，你如何来证明程序的性能好坏？ ---定义事务，或者该程序所在的文件的大小（如JS文件，文件越小越好）

25. 如果性能分析和调优应该是谁来负责？

26. 性能测试场景怎么搭建？

目录

[360笔试题 1](file:///C:\Users\Administrator\Desktop\360和百度测试笔试面试题.docx#_Toc428623298)

[360面试题 7](file:///C:\Users\Administrator\Desktop\360和百度测试笔试面试题.docx#_Toc428623299)

[百度一轮面试题 7](file:///C:\Users\Administrator\Desktop\360和百度测试笔试面试题.docx#_Toc428623300)

## 360笔试题

**测试标准：**

1. 编写测试计划的目的是：abc

A.使测试工作顺利进行 B.使项目参与人员沟通更加顺畅 C.使测试工作更加系统化条理化 D. 软件工程及软件过程的需要 E.软件过程规范化的要求 F.控制软件质量

2. 使用软件测试工具的目的是：e

A．帮助测试寻找问题 B.协助问题的诊断 C.节省测试时间 D.调高bug的发现率 E.更好的控制缺陷提高软件质量 F. 更好的协助开发人员

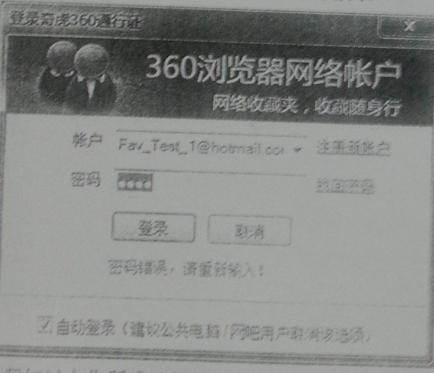
3. 测试结束的标准：bd

A．用例全部测试 B.覆盖率达到标准 C. 缺陷率达到标准 D. 其他指标达到质量标准

E.开发修复完了目前open的所有bug F. 下个产品的测试准备开始

**用例设计：**

1. 下方是奇虎通行证的登陆界面，输入正确的账户和密码后即可登陆，否则会提示不同的错误信息并阻止登陆，账户栏可接收邮箱或用户名，用户名可为中文或英文，最大长度为14个字符，请针对登陆这个功能写出测试用例



1. 假如以上你所涉及的某一个用例测出bug，请写出Bug报告

**网络、web篇：**

1. 什么是代理？代理是怎么工作的？
2. 什么是NAT
3. 写出windows下查看所有网络适配器配置的命令
4. 是否用过抓包工具？请写出你知道的抓包工具的名称。
5. 列出你所知道的常见的http状态码及所代表的含义。
6. http请求头通常包括哪些信息？
7. 什么是URL？请以下方这个URL为例说明URL每个组成部分的含义。

http://bbs.360.cn/3229787.html

1. 请简单描述一下，当用户单击一个按钮或在web页面上单击一个链接后，web系统响应的整个过程

**LINUX和UNIX篇**

写出完成以下任务的命令：

1. 查看当前所在目录。如若当前在，/home/usr1目录下，查看此目录大小。列出此目录下的所有文件（包括隐藏文件）。
2. 查看网络状态，查看网络适配器配置。查看系统进程，查看系统执行中的进程及其所占资源的命令。
3. 新建一个文件apptest.c
4. 列出所有你知道的可以用来查找文件和文件内容的命令。
5. 有300个源代码文件（.cpp , .h , .inl）放在同一个文件夹里面，由于版本升级，需要修改该文件夹中所有.cpp的文件中的版本号（由version=’1.2’更新为version=’3.0’）

要求：1.只修改.cpp文件中的版本号，不能影响.h和.inl文件 2.对原有的文件需要做备份，以.bak结尾

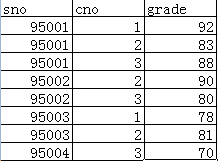
**前端篇：**

1. 在空白处补充js代码，取出input输入框的值<input id=’na’ />
2. 用html css js写一个页面，要求如下：页面有一个按钮，点击按钮后，在按钮的下方动态出现按钮的名字
3. 对js中的string对象进行扩展，使其具有去除前后空格的方法
4. 你认为如何提高网页的前端性能。

**SQL篇**

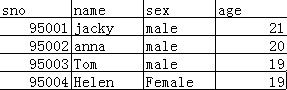
1. 有两个表，SC是学生选修课程成绩表，stu是学生信息表，请写出下面的SQL语句

表名：SC



sno：学号 cno: 课程编号 grade:分数

表名stu



1. 创建表stu，学号sno唯一且取值不能为空
2. 查询选修了课程的学生学号（去掉结果中的重复记录）
3. 查询所有选修了2号课程并且成绩在90以上的学生的姓名

**编程篇，以下C/C++，python，java，三选一**

**C、C++篇**

1. 请列举标准c/c++的基本变量类型（例如：int）,以及他们的字节长度。

2. 请用#define语句定义一个常量MSOFDAY，表示1天有多少毫秒

3. 有一段代码：

char \*pd=new char[32];

delete[] pd;

strcpy(pd,’abcd’);

char ps[32];

strcpy(ps,pd);

puts(ps);

请问程序能够打印出来‘abcd’串吗？请说明理由

4. 用c语言实现一个函数，将ASCII编码的大写字母转换成小写字母（‘A’=65,’a’=97）,

函数原型：

Int tolower(int c);

注意，不允许调用任何其他函数

**python篇**

1. 简述from……import……和import的区别

import a：

（1）导入a模块，所有的a模块的内容还在a模块的命名空间中，只是把对该模块的内存地址给引用到本地变量环境，所以引用a模块的变量都要加上a的模块名，比如a.var1

（2）一个模块多次重复使用import语句时，不会重新加载被指定的模块，只是把对该模块的内存地址给引用到本地变量环境

（3）重新import a的时候，a模块的代码不会执行（这是和reload的区别）

from a import \*：

（1）会把a模块的所有的顶层的变量、类导入到当前的命名空间中，所以使用a模块的变量的话不需要加前缀（这是和import的区别）

（2）模块只有第一次导入的时候，顶层代码会执行，再次导入的时候顶层代码不会执行，但是如果有同名参数的话会覆盖的

（3）"from 模块名 import \*"的副作用：如果有多个模块都用"from 模块名 import \*"导入，后面的引用就会把前面同名的变量覆盖

1. 简述字符串连接符+和join的区别，哪个更高效；
2. 举例说明range()和xgange()的用法及区别
3. 使用正则表达式匹配通讯录中的所有固定电话，例如010-58781234，输出所有匹配项（具体实现）
4. 对params\_dict={‘timestamp’:’133’,

‘merchant\_code’:’99’,

‘num’:’20120530’

‘notify’:’http://test/12312’,

‘return’:’http://test/12312’

‘product\_name’:’对流’，

‘ext’:’this is a demo’

}

进行字典排序

1. 请写出下面示例程序ll的运行结果

def f(n):

if n<1:

raise Exception

if n==1 or n==2:

return 1

return f(n-1)+f(n-2)

if \_name\_==’\_main\_’:

n=11

try:

print f(int(n))

except ValueError:

print ‘Please input a Integer’

except Exception:

print ‘Please input n=0’

1. 有一个urf8编码的文件a.txt，文件路径是E盘根目录，请写一段程序逐行读入这个文本文件，并在屏幕（GBK编码）上打印出来。
2. 编写一段程序用于发送Http协议的post请求，要求如下：

a.请求使用的url地址为：

<http://api.360.cn/intf.php?method=synoc.uploadFile&qid=21301377>

b.请求的data内容含两个键值对，分别是key=128211和local=hello\_360

c.请求头中需要包含cookie值为‘fname=hello360;’

d.需要连接代理192.168.33.81:5880发送请求

9. 解析如下config.xml文件，以key=value的形式存在字典里

<?xml version=’1.0’ encoding=’utf-8’?>

<config>

<db>

<host>192.168.1.1</host>

<use1>root</use1>

<password>root</password>

<dbname>mysql</dbname>

</db>

</config>

**java篇**

1. **基础题**
2. 列举一些常用的类，包，接口
3. 如何对一个HashMap按照字典序排序
4. 写出一个Singleton
5. **编程题**
6. 写出下面代码的结果：

public class TestString{

public static void main(String[] args){

String a=”Hello”;

linkText(a);

System.out.print(a);

}

public static void linkText(String a){

a+=”world”;

}

}

1. 写出下面代码的结果

public class Test{

pbblic static void main(String[] args){

String a=”ja”;

String b=”va”;

System.out.println(a+b=”java”);

System.out.println((a+b).equals(“java”));

}

}

**以下手机、性能、自动化的题目三选一**

**手机测试**

1. 手机测试需要哪些相关的通信知识（列出一些你所知道的即可）
2. 简述一下目前流行的移动平台有哪些，它们的特点是什么？在这些平台上做测试的手段方法上有什么不同？
3. 列举你所熟悉的手机操作系统中，有哪些特性、亮点让你印象深刻或者特别喜欢？
4. 手机测试都要考虑哪些性能测试点？
5. 列出你所知道的或者用过的能在手机测试上用到的工具（任何的）？
6. 请任意列举出3点你所熟悉的手机操作系统和app间交互的测试用例
7. 请写出Monkey测试中的最简单的执行命令
8. 手机平台实行自动化测试的原理有哪些，随便解释一种
9. 需要在android平台上抓取数据包进行测试，简述你的方法
10. Android调试工具是什么？列举几条简单的命令
11. 测试用例设计，请写出以下场景的详细测试步骤：

场景一：微博手机客户端中回复一条评论失败

场景二：私信中给用户A发送一条语音

**自动化测试篇**

1. 列出你所知道的自动测试工具和框架，选一个你所熟悉的，简述用它进行自动化测试的步骤
2. 什么样的产品或模块适合实施自动化测试？

**QTP篇**

1. GetROProperty,GetTOProperty,GetTOProperties 3个函数的区别
2. 什么是检查点，检查点的作用是什么，请列举出你所知道的检查点类型
3. 在VBScript中，正则表达式通常被应用到什么地方，并且写出一个验证有效Email地址的正则表达式。
4. 用VBScript删除E盘下的test文件夹
5. 写一小段QTP脚本，需要将global sheet中testvalue列第2行到第5行的数据读取出，读取成功后打印这些数据，请用testinput(testvalue)函数

**selenium篇**

1. webdriver与selenium的区别
2. 断言(assert)和验证（verify）的区别，分别何时使用
3. selenium如何测试ajax功能，简述即可

**自动化实战篇**

将如下的测试场景用最熟悉的自动化工具实现，写出具体的实施步骤和脚本内容，测试数据自己准备

测试内容：上传功能

测试目的：上传功能可支持一次上传999个文件

测试步骤：1.准备数据 2.上传文件 3.等待上传完成 4.判断上传完成及成功预期结果：上传成功

说明:每个文件内容需不一样，文件名无要求

**性能篇**

1. 什么是性能测试，列出常用的性能指标
2. 列出你所知道的性能测试工具

**LR篇**

1. LR的主要组成部件是什么？主要作用分别是什么
2. 什么是集合点?LR中设置集合点的函数是哪个？
3. 用什么函数可以将脚本中自定义的参数转化成LR的参数
4. 关联是做什么用的？手动关联的主要步骤是什么且用到的最关键的是哪个函数
5. 如何在Controllor里添加windows和linux资源监控
6. 解释以下函数及他们的不同之处：

lr\_debug\_message

lr\_output\_message

lr\_error\_message

lrd\_stmt

lrd\_fetch

1. lR调用动态链接库的步骤是什么

## 360面试题

1. 简述一个工作中项目，测试过程，测试周期，以及项目中你充当的角色
2. 自动化、性能测试实施过程
3. 做性能测试时，监控服务器的命令。
4. cpu、io、内存的关系，如果cpu低，io高，内存高，分析问题出在哪儿，为什么？
5. 分析性能问题的思路是什么
6. 压力测试、负载测试、稳定测试分别是什么
7. 如何用LR对登陆功能做性能测试，简述实施过程和步骤，如果单次执行报错，你认为是什么错误。
8. LR性能脚本里有动态参数，用什么方法解决，如何获取动态参数，具体步骤是什么，用到的函数是什么
9. 检查点的函数是什么
10. 性能测试指标是什么
11. 如果不优化程序，不增加机器，如何提高系统性能？
12. 程序题:逐行扫描文本文件，找出每行中只包含字母的字符串，并输出

例如：123abc456bcd，输出这行中的abc，bcd

1. 说出http你知道的状态码，分别是什么含义
2. 用过接口测试工具是什么，抓包工具用过什么
3. 时间复杂度是什么，Python中，获取字典中key的时间复杂度
4. 程序如何实现时间复杂度为O（1）
5. python中，列表和元组的区别
6. 性能测试中，如果用100个并发用户做压力测试，tps会达到100以上吗？为什么
7. vuser、tps、qps是什么，他们的关系和区别是什么
8. 性能测试中，根据什么数据分析最优点
9. 对以往的工作，如果再让你做一次，你认为有什么改进之处
10. 印象最深刻的Bug是什么
11. 工作中和生活中，别人对你的评价，用三个词描述

## 百度一轮面试题

1. 针对三角形写出测试用例，判断它是不是三角形，是不是等边、等腰三角形
2. 如何判断单链表中是否有环，有没有高效的方法
3. 冒泡排序、快排是什么，请写出冒泡排序的代码
4. java中，Int和integer的区别
5. 用Linux命令，删除文件中含有abc的那一行
6. 用linux命令，统计文件中所有的IP出现次数，显示出现次数最多的前5名
7. 数据库:将一个表按age排序，取出第二大的age值
8. 如何备份数据库，如何恢复，写出具体命令
9. 说出你知道的数据库join查询方式，分别是什么意思
10. 是否会c和C++
11. 简述性能测试实施过程，分析哪些性能数据，如何导出linux服务器中的cpu，io，内存性能测试数据
12. 如何看数据库是否有慢查询

## 猎豹面试

1. 链表概念，特点?
2. 以5只猩猩5分钟可以吃5支香蕉速度计算，要在100分钟内吃100支香蕉，需要多少只猩猩？
3. Linux服务和进程管理：

A.Linux的启动过程简单描述为五个阶段，主机加电启动；加载引导程序；加载ini进程，加载启动Linux内核；运行终端和验证用户登录

B.Linux系统中，/etc/rc.local文件的脚本或命令，开机后会被自动执行

C.Linux的crontab可以计划调用任务在执行时间执行，其表达方式为hour minute day month day of the week command

D.head命令可以监控执行用户产生的进程，按下u健，然后输入要监视的用户即可

1. 描述以下表示形式的意义：Char \*q[]={“xxx”，”yyy”, “zzz”}; Char (\* p) []=a;

A.数组、指针数组 B.数组指针、指针数组 C.指针致组、数组指针；D\_数组、数组指针

1. 下列哪个命令是正确的，查找/etc/sysctl.conf文件属于哪个软件包？

A. rpm -qi /etc/sysctl. conf B. rpm -qa /etc/sysctl. conf,

C. rpm -ql / etc/sysctl. conf, D. rpm -qf / etc/sysct1. conf

1. 甲、乙两人同时从A、B两地出发相向而行，两人在距离A地40米处第一次相遇。相遇后两人仍以原速继续前行，并且各自到达对方出发点后立即沿原路返回，途中两人在距离B地15米处第二次相遇。A、B两地相距多少米？
2. 如果将网络IP段40.15.128.0/17划分成两个子网，则第一个子网IP段为40.15,128.0/18，则第二个子网为?

A. 40. 15. 129. 0/18 B. 40. 15. 128. 128/18

C. 40. T5. 192. 0/I7 D. 40, 15. 192. 0/18

1. 完全二叉树，叶节点？
2. 整型在16位系统占2字节，在32位系统中占4字节，另外也会根编译器有关。
3. Java中方法重写与重载的区别？
4. 边界值测试方法的优缺点？
5. 请写出sql注入攻击或xss攻击语句？
6. 请简述进程死锁的产生条件以及发现此类问题的测试手段？
7. 请简述接口测试方法以及注意事项？

## 爱投资测试开发

1. **Linux使用&shell**
2. 将当前目录下的名为img的文件夹拷贝到当前路径下的static文件夹中\_\_\_\_\_\_\_\_\_\_删除当前路径下的static文件夹\_\_\_\_\_\_\_\_\_
3. 动态监控nginx.log的新增内容，输出其中含有"[error]"字符串的行\_\_\_\_\_\_\_\_\_\_
4. 用vim打开文本文件config.php后，如何将所有的字符串"true"替换为"false"\_\_\_\_\_\_\_\_如何加行号\_\_\_\_\_\_\_\_\_如何删除光标所在行起始的5行\_\_\_\_\_\_\_如何撤销上一步操作\_\_\_\_\_如何直接到达文件的最后一行\_\_\_\_\_\_\_\_
5. 不使用搜索引擎，在Linux命令行下，如何查看某个命令如何使用（即Linux的手册命令）\_\_\_\_\_\_\_
6. shell编程题，用bash实现如下功能
7. 每隔1分钟，在当前文件夹下生成一个以当前日期加顺序号为名字的文件夹，如使用相关Linux命令获得了今天日期是20150402，则生成的文件夹依次是20150402\_1,20150402\_2,20150402\_3等，依次类推；
8. 上述问题条件改为每隔10分钟，其他要求不变，请再次实现该功；
9. **数据库**

假设有张学生成绩表（表名：table\_grade）如下

name subject grade

张三 语文 73

张三 数学 83

张三 物理 93

李四 语文 74

李四 数学 84

李四 物理 94

1. 编写SQL语句，实现如下功能
2. 输出语文这门课成绩不低于60分的人数
3. 输出数学这门课成绩前5名的平均分
4. 输出两列，第一列是课程名，第二列是该课程名的平均分
5. 输出重名的名字以及该名字重名的次数，即输出两列，第一列是重名的名字，第二列是其重复的次数
6. select \* from table\_grade where grade in(90, 91, 92, 93);这条sql在grade字段没有索引的情况下时间复杂度是多少？grade字段有索引时，时间复杂度是多少？
7. 了解或使用过哪些非关系型数据库（NoSQL）?
8. **网络&性能**
9. 查看CPU idle的命令\_\_\_\_\_\_\_\_查看Linux服务器磁盘空间占用情况的命令是\_\_\_\_\_\_\_
10. 自下而上的网络五层结构分别是\_\_\_\_\_\_、\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_FTP、PSPS、STMP是\_\_\_\_\_\_\_层的协议，端口是\_\_\_\_\_层的概念，mac地址是什么层的概念\_\_\_\_\_\_
11. netstat -n | grep ^tcp | awk '{print $NF}' | sort -nr | uniq -c，该Linux命令的作用是什么？
12. 写出常见的HTTP状态码及其含义
13. TCP连接中有若干状态，写出ESTABLISHED、TIME\_WAIT、CLOSE\_WAIT这三个状态对应的含义
14. 了解或使用过哪些压力测试工具
15. **编程题**

说明：以下四题任选一题编程实现，前两题不限语言（C/C++/JAVA/PHP等皆可）

1. 编写BubbleSort函数，实现对数组的冒泡排序
2. 编写ReverseList函数，实现单向链表的逆置操作；其中对于链表的节点的定义，如若使用C/C++语言，可以定义如下：

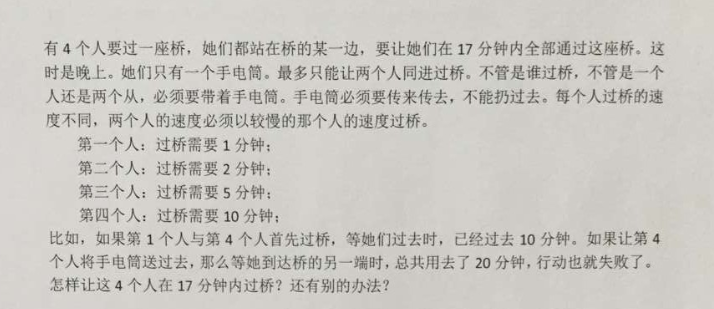
typedef struct node{

int data;

struct node \*next;

}Node;

1. 实现C语言中的strcmp函数，即字符串比较功能，如果两字符串相等，返回0，否则返回非0；
2. 实现C语言中的strstr函数，即在字符串中查找字字符串，若找不到子串，返回NULL；若能找到则返回指向该子串的指针；
3. **Python题**
4. Python中pass语句的作用是什么
5. Python里面如何生成随机数
6. 给定dict={'a' : 3, 'bc' : 5, 'c' : 3, 'asd' : 4, '33' : 56, 'd' : 0}，根据其键值进行排序
7. 实现删除a = [1, 3, 2, 2, 1, 5, 5, 3]里面的重复元素，有几种方法，并写出来
8. 输入三个整数x, y, z，请把这三个数由小到大输出
9. 简单介绍一下你之前的工作，你对你之前的工作有什么评价
10. 你为什么选择做测试
11. 测试的流程
12. 设计用例有哪些方法，并说说其含义
13. 你认为做好测试用例设计工作的关键是什么
14. 怎么做接口测试，后台如何辨别接口中传入的用户就是你使用的用户而不是别人冒充你的用户
15. 接口传给你的是一些什么样的数据，在传递数据的过程中需要注意什么
16. 给你一个登录框，请设计用例，是否允许无限次错误登录，如不允许如何处理
17. 给你一个没有需求的系统，你如何测试
18. http协议中的get、post有什么区别，使用post方式数据是如何传递的
19. session和cookie的区别
20. 你知道哪些排序算法，其核心思想是什么，并写出一种排序算法（冒泡写的最多）
21. 你觉得你写的排序算法有哪些缺陷，并进行用例设计
22. 介绍一下你知道的一些数据结构（堆、栈、队列等）
23. 链表和数组，哪个索引数据的效率高
24. 你做过哪些安全方面的测试，用过哪些安全扫描工具，并简单介绍一下
25. Linux常用命令及用法（特别是find、grep、awk，更多的是现场写组合的Linux命令语句）
26. mysql有哪些存储引擎，它们之间的区别是什么？
27. 什么叫事务，什么叫锁，mysql存在哪些锁
28. 索引是不是越多越好，如果不是，那什么情况下不易创建过多的索引
29. 获取数据表中某一天的数据（时间不定），你有哪些方法
30. Python中列表和元组的区别
31. 你用Python写过哪些程序，简单介绍一下
32. 你知道Python的哪些模块，怎么解析json
33. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别
34. 你怎样规划你的职业生涯



P2P P2C 白条（金融上）

## 新东方

* 怎么做好文档测试？
* 白盒测试有几种方法？
* Alpha测试与beta测试的区别？
* 比较性能测试与压力测试的区别？
* Session与cookie的区别？
* 你认为做好测试用例设计工作的关键是什么？
* 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别和联系？

## 阿里面试

1、如何理解自动化测试，用测试工具进行测试等于自动化测试这句话对不对？

　　关注点：测试工具的使用是自动化测试的一部分工作，但“用测试工具进行测试”不等于“自动化测试”。自动化测试，模拟手工测试步骤，通过执行程序语言编制的 测试脚本自动地测试软件。 自动化测试，强调借助工具（不仅仅是工具，有时包括策略和工件）来完成测试的执行，也就是用工具来帮助或辅助测试。但是用测试工具进行测试有可能是自动化，半自动化，或者手工测试。

　　2、介绍下比较了解的自动化框架，watir，selenium，QTP…..任选一个说说，这个框架的工作原理是什么？

　　随便选取一个，重要的是原理，而不是使用。大家在用这些框架的时候，一定要关注背后的执行原理.看源码是一个比较简单的途径。

　　3、介绍下SoapUI，如果你用着的话。这个框架需要注意什么？

　　soapUI是一款桌面应用程序，能够监测、触发、模仿以及测试（功能和负载）基于SOAP/WSDL和REST/EADL的HTTP网络服务。

　　和大多数的工具一样，都是使用HTTPREQUEST对相应的资源进行请求很提取。再得到response之后进行相应的处理，对XML进行XPATH定位。注意的是SOAP方法中包含GET，POST的方法，POST的方法主要使用Application/xml的MIME形式发送相应的POST数据。

　　4、对webservice层面的自动化测试，你认为比较重要的是什么？

　　对webservice的测试主要分为两个阶段，首先是对WEB Ui层面的数据XML Response与webservice的schema进行对比测试，其次是web Ui层面的数据与数据库服务器中相应的数据进行验证。

　　5、对持续集成工具有了解过吗？类似于Jenkins（hudsoon）/Bamboo/Teamcity这些持续集成的工具，有了解过这些吗？

　　目前比较这几个还算比较流行，阿里主要集中在用hudson。Teamcity在以前的公司了解过。

　　6、桌面自动化测试和WEB 自动化测试的区别？

　　驱动方式不同，C/S架构（或者桌面类型）界面自动化测试，采取的方式可以调用操作系统本身的API（windows桌面软件）来构建自动化测试或者可以采用虚拟机内（java swing程序）的事件处理机制来完成了。

　　WEB 自动化测试 B/S架构，原理就是依靠JS来进行客户端的操作，然后寻找对象是采用了DOM解析技术，将web方面的节点进行解析定位

　　7、自动化测试碰到比较难解决的问题是什么？如果出现这些问题给出你的解决方案？

　　重点引导到测试结果定位准确这个角度上来， 在自动化程度比较高，case很多，就会存在排查失败的case过程。

　　解决方案； case错误分类，有效的log日志，异常信息的抓取

　　8、IOS支持UI自动化，主要有2种方式，介绍下这2种方式？

　　1）苹果官方提供的技术， UI Automation。

　　2）就是在应用中注入测试代码。

Instrument uiautomation 是苹果官方提供的iPhone手机应用的自动化测试工具。控件元素的识别准确，属性获取，元素操作的API丰富。可以很方便的录制测试脚本、回放和查看运行结果。

面试题第一部分

1、什么是兼容性测试？兼容性测试侧重哪些方面？

参考答案：

兼容测试主要是检查软件在不同的硬件平台、软件平台上是否可以正常的运行，即是通常说的软件的可移植性。

兼容的类型，如果细分的话，有平台的兼容，网络兼容，数据库兼容，以及数据格式的兼容。

兼容测试的重点是，对兼容环境的分析。通常，是在运行软件的环境不是很确定的情况下，才需要做兼容。根据软件运行的需要，或者根据需求文档，一般都能够得出用户会在什么环境下使用该软件，把这些环境整理成表单，就得出做兼容测试的兼容环境了。

兼容和配置测试的区别在于，做配置测试通常不是Clean OS下做测试，而兼容测试多是在Clean OS的环境下做的。

2、我现在有个程序，发现在Windows上运行得很慢，怎么判别是程序存在问题还是软硬件系统存在问题？

参考答案：

1、检查系统是否有中毒的特征；

2、检查软件/硬件的配置是否符合软件的推荐标准；

3、确认当前的系统是否是独立，即没有对外提供什么消耗CPU资源的服务；

4、如果是C/S或者B/S结构的软件，需要检查是不是因为与服务器的连接有问题，或者访问有问题造成的；

5、在系统没有任何负载的情况下，查看性能监视器，确认应用程序对CPU/内存的访问情况。

3、测试的策略有哪些？

参考答案：

黑盒/白盒，静态/动态，手工/自动，冒烟测试，回归测试，公测（Beta测试的策略）

4、正交表测试用例设计方法的特点是什么？

参考答案：

用最少的实验覆盖最多的操作，测试用例设计很少，效率高，但是很复杂；

对于基本的验证功能，以及二次集成引起的缺陷，一般都能找出来；但是更深的缺陷，更复杂的缺陷，还是无能为力的；

具体的环境下，正交表一般都很难做的。大多数，只在系统测试的时候使用此方法。

5、描述使用bugzilla缺陷管理工具对软件缺陷（BUG）跟踪的管理的流程？

参考答案：

就是Bugzilla的状态转换图。

6、你觉得bugzilla在使用的过程中，有什么问题？

参考答案：

界面不稳定；

根据需要配置它的不同的部分，过程很烦琐。

流程控制上，安全性不好界定，很容易对他人的Bug进行误操作；

没有综合的评分指标，不好确认修复的优先级别。

7、描述测试用例设计的完整过程？

参考答案：

需求分析 + 需求变更的维护工作；

根据需求 得出测试需求；

设计测试方案，评审测试方案；

方案评审通过后，设计测试用例，再对测试用例进行评审；

8、单元测试的策略有哪些？

参考答案：

逻辑覆盖、循环覆盖、同行评审、桌前检查、代码走查、代码评审、景泰数据流分析

9、LoadRunner分哪三部分？

参考答案：

用户动作设计；

场景设计；

测试数据分析；

10、LoadRunner进行测试的流程?

参考答案：

1、 测试测试

2、 创建虚拟用户脚本

3、 创建运行场景

4、 运行测试脚本

5、 监视场景

6、 分析测试的结果

以上，最好是结合一个案例，根据以上流程来介绍。

什么是并发？在lordrunner中，如何进行并发的测试？集合点失败了会怎么样？

参考答案：

在同一时间点，支持多个不同的操作。

LoadRunner中提供IP伪装，集合点，配合虚拟用户的设计，以及在多台电脑上设置，可以比较好的模拟真实的并发。

集合点，即是多个用户在某个时刻，某个特定的环境下同时进行虚拟用户的操作的。集合点失败，则集合点的才操作就会取消，测试就不能进行。

12、使用QTP做功能测试，录制脚本的时候，要验证多个用户的登录情况/查询情况，如何操作？

参考答案：

分析用户登录的基本情况，得出一组数据，通过性测试/失败性测试的都有（根据TC来设计这些数据），然后录制登录的脚本，将关键的数据参数化，修改脚本，对代码进行加强，调试脚本。

13、QTP中的Action有什么作用？有几种？

参考答案：

Action的作用

n 用Action可以对步骤集进行分组

n 步骤重组，然后被整体调用

n 拥有自己的sheet

n 组合有相同需求的步骤，整体操作

n 具有独立的对象仓库

Action的种类

n 可复用Action

n 不可复用Action

n 外部Action

14、TestDirector有些什么功能，如何对软件测试过程进行管理？

参考答案：

需求管理

n 定义测试范围

n 定义需求树

n 描述需求树的功能点

测试计划

n 定义测试目标和测试策略。

n 分解应用程序，建立测试计划树。

n 确定每个功能点的测试方法。

n 将每个功能点连接到需求上，使测试计划覆盖全部的测试需求。

n 描述手工测试的测试步骤

n 指明需要进行自动测试的功能点

测试执行

n 定义测试集合。

n 为每个测试人员制定测试任务和测试日程安排。

n 运行自动测试。

缺陷跟踪

n 记录缺陷

n 查看新增缺陷，并确定哪些是需要修正的

n 相关技术人员修改缺陷

n 回归测试

n 分析缺陷统计图表，分析应用程序的开发质量。

15、你所熟悉的软件测试类型都有哪些？请试着分别比较这些不同的测试类型的区别与联系（如功能测试、性能测试……）？

参考答案：Compatibility Testing（兼容性测试），也称“Configuration testing（配置测试）”，测试软件是否和系统的其它与之交互的元素之间兼容，如：浏览器、操作系统、硬件等。验证测试对象在不同的软件和硬件配置中的运行情况。

Functional testing (功能测试)，也称为behavioral testing（行为测试），根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。

Performance testing（性能测试），评价一个产品或组件与性能需求是否符合的测试。包括负载测试、强度测试、数据库容量测试、基准测试等类型。

16、软件缺陷（或者叫Bug）记录都包含了哪些内容？如何提交高质量的软件缺陷（Bug）记录？

参考答案：5C标准

17、Beta测试与Alpha测试有什么区别？

参考答案：Beta testing(β测试),测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场

Alpha testing (α测试),是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试

18、软件的评审一般由哪些人参加？其目的是什么？

参考答案：

在正式的会议上将软件项目的成果（包括各阶段的文档、产生的代码等）提交给用户、客户或有关部门人员对软件产品进行评审和批准。其目的是找出可能影响软件产品质量、开发过程、维护工作的适用性和环境方面的设计缺陷，并采取补救措施，以及找出在性能、安全性和经济方面的可能的改进。

人员：用户、客户或有关部门开发人员，测试人员，需求分析师都可以，就看处于评审那个阶段

19、测试活动中，如果发现需求文档不完善或者不准确，怎么处理？

参考答案：

测试需求分析发现需求文档不完善或者不准确，应该立即和相关人员进行协调交流。

20、阶段评审与项目评审有什么区别？

参考答案：

阶段评审对项目各阶段评审：对阶段成果和工作

项目评审对项目总体评审：对工作和产品

21、阐述工作版本的定义？

参考答案：

构造号： BUILD

22、什么是桩模块？什么是驱动模块？

参考答案：

桩模块：被测模块调用模块

驱动模块调用被测模块

23、什么是扇入？什么是扇出？

参考答案：

扇入：被调次数，扇出：调其它模块数目

24、你认为做好测试计划工作的关键是什么？

参考答案：

软件测试计划就是在软件测试工作正式实施之前明确测试的对象，并且通过对资源、时间、风险、测试范围和预算等方面的综合分析和规划，保证有效的实施软件测试；

做好测试计划工作的关键：目的，管理，规范

1. 明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

2．坚持“5W”规则，明确内容与过程

“5W”规则指的是“What（做什么）”、“Why（为什么做）”、“When（何时做）”、“Where（在哪里）”、“How（如何做）”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的（Why），明确测试的范围和内容（What），确定测试的开始和结束日期（When），指出测试的方法和工具（How），给出测试文档和软件的存放位置（Where）。

3．采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

4. 分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

25、你认为做好测试用例工作的关键是什么？

参考答案：

需求和设计文档的理解程度，对系统的熟悉程度

26、简述一下缺陷的生命周期？

参考答案：提交->确认->分配->修复->验证->关闭

27、软件的安全性应从哪几个方面去测试？

参考答案：

(1)用户认证机制：如数据证书、智能卡、双重认证、安全电子交易协议

(2)加密机制

(3)安全防护策略：如安全日志、入侵检测、隔离防护、漏洞扫描

(4)数据备份与恢复手段：存储设备、存储优化、存储保护、存储管理

(5)防病毒系统

28、软件配置管理工作开展的情况和认识？

参考答案：

软件配置管理贯穿于软件开发、测试活动的始终，覆盖了开发、测试活动的各个环节，它的重要作用之一就是要全面的管理保存各个配置项，监控各配置项的状态，并向项目经理及相关的人员报告，从而实现对软件过程的控制。

软件测试配置管理包括4个最基本的活动：

配置项标识

配置项控制

配置项状态报告

配置审计

软件配置管理通常借助工具来辅助，主要有MS SourceSafe、Rational ClearCase等

29、你觉得软件测试通过的标准应该是什么样的？

参考答案：

缺陷密度值达到客户的要求

30、引入测试管理的含义？

参考答案：风险分析，进度控制、角色分配、质量控制

31、一套完整的测试应该由哪些阶段组成？

参考答案：测试计划、测试设计与开发、测试实施、测试评审与测试结论

32、单元测试的主要内容？

参考答案：

　模块接口测试、局部数据结构测试、路径测试、错误处理测试、边界测试

33、集成测试也叫组装测试或者联合测试，请简述集成测试的主要内容？

参考答案：

（1）在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失；

　（2）一个模块的功能是否会对另一个模块的功能产生不利的影响；

　（3）各个子功能组合起来，能否达到预期要求的父功能；

　（4）全局数据结构是否有问题；

　（5）单个模块的误差累积起来，是否会放大，从而达到不能接受的程度。

34、简述集成测试与系统测试关系？

参考答案：

　（1）集成测试的主要依据概要设计说明书，系统测试的主要依据是需求设计说明书；

　（2）集成测试是系统模块的测试，系统测试是对整个系统的测试，包括相关的软硬件平台、网络以及相关外设的测试。

35、软件测试的文档测试应当贯穿于软件生命周期的全过程，其中用户文档是文档测试的重点。那么软件系统的用户文档包括哪些？

参考答案：

　　用户手册

　　安装和设置指导

　　联机帮助

　　指南、向导

　　样例、示例和模板

　　授权/注册登记表

最终用户许可协议

36、软件系统中除用户文档之外，文档测试还应该关注哪些文档？

参考答案：

开发文档

软件需求说明书

　　　　数据库设计说明书

　　　　概要设计说明书

　　　　详细设计说明书

　　　　可行性研究报告

管理文档

　　　　项目开发计划

　　　　测试计划

　　　　测试报告

　　　　开发进度月报

　　　　开发总结报告

37、简述软件系统中用户文档的测试要点？

参考答案：

　（1）读者群。文档面向的读者定位要明确。对于初级用户、中级用户以及高级用户应该有不同的定位

　（2）术语。文档中用到的术语要适用与定位的读者群，用法一致，标准定义与业界规范相吻合。

　（3）正确性。测试中需检查所有信息是否真实正确，查找由于过期产品说明书和销售人员夸大事实而导致的错误。检查所有的目录、索引和章节引用是否已更新，尝试链接是否准确，产品支持电话、地址和邮政编码是否正确。

　（4）完整性。对照软件界面检查是否有重要的分支没有描述到，甚至是否有整个大模块没有描述到。

　（5）一致性。按照文档描述的操作执行后，检查软件返回的结果是否与文档描述的相同。

　（6）易用性。对关键步骤以粗体或背景色给用户以提示，合理的页面布局、适量的图表都可以给用户更高的易用性。需要注意的是文档要有助于用户排除错误。不但描述正确操作，也要描述错误处理办法。文档对于用户看到的错误信息应当有更详细的文档解释。

　（7）图表与界面截图。检查所有图表与界面截图是否与发行版本相同。

　（8）样例与示例。像用户一样载入和使用样例。如果是一段程序，就输入数据并执行它。以每一个模块制作文件，确认它们的正确性。

　（9）语言。不出现错别字，不要出现有二义性的说法。特别要注意的是屏幕截图或绘制图形中的文字。

　（10）印刷与包装。检查印刷质量；手册厚度与开本是否合适；包装盒的大小是否合适；有没有零碎易丢失的小部件等等。

38、单元测试主要内容是什么？

参考答案：

单元测试大多数由开发人员来完成，测试人员技术背景较好或者开发系统软件时可能会安排测试人员进行单元测试，大多数进行的单元测试都是开发人员调试程序或者开发组系统联合调试的过程。讨论这个问题主要是扩充一下读者的视野。

单元测试一般包括五个方面的测试：

（1）模块接口测试：模块接口测试是单元测试的基础。只有在数据能正确流入、流出模块的前提下，其他测试才有意义。模块接口测试也是集成测试的重点，这里进行的测试主要是为后面打好基础。测试接口正确与否应该考虑下列因素：

-输入的实际参数与形式参数的个数是否相同；

-输入的实际参数与形式参数的属性是否匹配；

-输入的实际参数与形式参数的量纲是否一致；

-调用其他模块时所给实际参数的个数是否与被调模块的形参个数相同；

-调用其他模块时所给实际参数的属性是否与被调模块的形参属性匹配；

-调用其他模块时所给实际参数的量纲是否与被调模块的形参量纲一致；

-调用预定义函数时所用参数的个数、属性和次序是否正确；

-是否存在与当前入口点无关的参数引用；

-是否修改了只读型参数；

-对全程变量的定义各模块是否一致；

-是否把某些约束作为参数传递。

如果模块功能包括外部输入输出，还应该考虑下列因素：

-文件属性是否正确；

-OPEN/CLOSE语句是否正确；

-格式说明与输入输出语句是否匹配；

-缓冲区大小与记录长度是否匹配；

-文件使用前是否已经打开；

-是否处理了文件尾；

-是否处理了输入/输出错误；

-输出信息中是否有文字性错误。

-局部数据结构测试；

-边界条件测试；

-模块中所有独立执行通路测试；

（2）局部数据结构测试：检查局部数据结构是为了保证临时存储在模块内的数据在程序执行过程中完整、正确，局部功能是整个功能运行的基础。重点是一些函数是否正确执行，内部是否运行正确。局部数据结构往往是错误的根源，应仔细设计测试用例，力求发现下面几类错误：

-不合适或不相容的类型说明；

-变量无初值；

-变量初始化或省缺值有错；

-不正确的变量名（拼错或不正确地截断）；

-出现上溢、下溢和地址异常。

（3）边界条件测试：边界条件测试是单元测试中最重要的一项任务。众所周知，软件经常在边界上失效，采用边界值分析技术，针对边界值及其左、右设计测试用例，很有可能发现新的错误。边界条件测试是一项基础测试，也是后面系统测试中的功能测试的重点，边界测试执行的较好，可以大大提高程序健壮性。

（4）模块中所有独立路径测试：在模块中应对每一条独立执行路径进行测试，单元测试的基本任务是保证模块中每条语句至少执行一次。测试目的主要是为了发现因错误计算、不正确的比较和不适当的控制流造成的错误。具体做法就是程序员逐条调试语句。常见的错误包括：

-误解或用错了算符优先级；

-混合类型运算；

-变量初值错；

-精度不够；

-表达式符号错。

比较判断与控制流常常紧密相关，测试时注意下列错误：

-不同数据类型的对象之间进行比较；

-错误地使用逻辑运算符或优先级；

-因计算机表示的局限性，期望理论上相等而实际上不相等的两个量相等；

-比较运算或变量出错；

-循环终止条件或不可能出现；

-迭代发散时不能退出；

-错误地修改了循环变量。

模块的各条错误处理通路测试：程序在遇到异常情况时不应该退出，好的程序应能预见各种出错条件，并预设各种出错处理通路。如果用户不按照正常操作，程序就退出或者停止工作，实际上也是一种缺陷，因此单元测试要测试各种错误处理路径。一般这种测试着重检查下列问题：

-输出的出错信息难以理解；

-记录的错误与实际遇到的错误不相符；

-在程序自定义的出错处理段运行之前，系统已介入；

-异常处理不当；

-错误陈述中未能提供足够的定位出错信息。

39、如何理解强度测试？

参考答案：

强度测试是为了确定系统在最差工作环境的工作能力,也可能是用于验证在标准工作压力下的各种资源的最下限指标。

它和压力测试的目标是不同的,压力测试是在标准工作环境下,不断增加系统负荷,最终测试出该系统能力达到的最大负荷(稳定和峰值),而强度测试则是在非标准工作环境下,甚至不断人为降低系统工作环境所需要的资源,如网络带宽,系统内存,数据锁等等,以测试系统在资源不足的情况下的工作状态,通过强度测试,可以确定本系统正常工作的最差环境.

强度测试和压力测试的测试指标相近,大多都是与时间相关的指标,如并发量(吞吐量),延迟(最大\最小\平均)以及顺序指标等

强度测试需要对系统的结构熟悉,针对系统的特征设计强度测试的方法

40、如何理解压力、负载、性能测试测试？

参考答案：

性能测试是一个较大的范围，实际上性能测试本身包含了性能、强度、压力、负载等多方面的测试内容。

压力测试是对服务器的稳定性以及负载能力等方面的测试，是一种很平常的测试。增大访问系统的用户数量、或者几个用户进行大数据量操作都是压力测试。而负载测试是压力相对较大的测试，主要是测试系统在一种或者集中极限条件下的相应能力，是性能测试的重要部分。100个用户对系统进行连续半个小时的访问可以看作压力测试，那么连续访问8个小时就可以认为负载测试，1000个用户连续访问系统1个小时也可以看作是负载测试。

实际上压力测试和负载测试没有明显的区分。测试人员应该站在关注整体性能的高度上来对系统进行测试。

有1000瓶水，其中有一瓶有毒，小白鼠只要尝一点带毒的水24小时后就会死亡，至少要多少只小白

时间不是问题，24小时内肯定可以找出有毒的那瓶。

给1000个瓶分别标上如下标签（10位长度）：

0000000001 （第1瓶）

0000000010 （第2瓶）

0000000011 （第3瓶）

......

1111101000 （第1000瓶）

从编号最后1位是1的所有的瓶子里面取出1滴混在一起（比如从第一瓶，第三瓶，。。。

里分别取出一滴混在一起）并标上记号为1。以此类推，从编号第一位是1的所有的瓶子

里面取出1滴混在一起并标上记号为10。现在得到有10个编号的混合液，小白鼠排排站，

分别标上10，9，。。。1号，并分别给它们灌上对应号码的混合液。24小时过去了，过来验尸吧：

从左到右，死了的小白鼠贴上标签1，没死的贴上0，最后得到一个序号，把这个序号换成10进制的数字，就是有毒的那瓶水的编号。

检验一下：假如第一瓶有毒，按照0000000001 （第1瓶），说明第1号混合液有毒，因此小白鼠的生死符为0000000001（编号为1的小白鼠挂了），0000000001二进制标签转换成十进制=1号瓶有毒；假如第三瓶有毒，0000000011 （第3瓶），第1号和第2号混合液有毒，因此小白鼠的生死符为00000011（编号为1，2的鼠兄弟挂了），0000000011二进制标签转换成十进制=3号瓶有毒。

记一次性能测试实践

1.测试对象

这次测了一些http接口和几个网页。

2.测试策略

2.1 基准测试：单个调用各接口循环100次计算平均响应时间

2.2 性能测试：单个接口调用以50并发用户数为单位，逐步加压直到预估的实际负载300并发用户，观察测试指标变化

2.3 压力测试：单个接口调用以50并发用户数为单位，逐步加压直到错误率过高或服务器资源使用率过高，观察测试指标变化

2.4 负载测试：预估实际负载为300并发用户数，在此基础上持续测试5分钟左右，观察测试指标是否达标

2.5 稳定性测试：预估实际负载为300并发用户数，在此基础上持续测试60分钟左右，观察测试指标是否达标，重点观察错误率

2.6 疲劳性测试：预估实际负载为300并发用户数，在此基础上持续测试240分钟左右，观察测试指标是否达标，重点观察错误率

2.7 组合测试：对2.2-2.5的测试采用不同接口同时调用（即系统不同模块同时测试）

2.8 其他：以不同ip地址加压，测试服务器负载均衡效果。

以上，本次只做了2.2、2.3、2.4、2.8

3.测试指标

测响应时间、错误率；同时专人监控服务器硬件资源使用状况、监控tomcat应用服务器等。

计算和监控吞吐量（测试工具自动计算测试执行过程中的吞吐量（每秒钟处理请求数），同时服务器监控软件业监控到了测试执行时服务器的吞吐量）

本次实际测试得到吞吐量距离预估有较大差距；错误率超出预期；且测试数据准备有一定问题。

4.测试工具

本次选用Jmeter，因为便宜且灵活。

需设置语言为英文，默认中文翻译不完整。

5.测试脚本编写、调试

5.1 提前对接口、网页进行录制。每个待测接口、网页需要加断言。 断言多采用JQuery断言和Regular Expression断言

5.2 重点在测试数据的准备。

5.3 采用了本地web应用提供数据，jmeter获取这些数据，再发送给服务器的方法（这次发现这个本地应用生成的数据在较高并发时有重复，导致了不必要的错误率）

5.4 测试结果监听器： assertion results, summary report, aggregate report, result tree, result table

5.5 测试接口调用时，可用网页、数据库等其他方法确认接口调用成功。观察接口调用是否生效，是否和网页同样效果。

6.测试执行

6.1 一台电脑加压300-600并发用户。如果需要更多则需要增加电脑。

6.2 以不同ip地址加压，测试服务器负载均衡效果。

6.3 机房测试，排除internet网络延迟问题

6.4 数据备份和还原，排除性能测试对数据的改变

6.5 生产环境测试（系统未上线），排除测试环境的影响

7.测试报告

7.1 截取了jmeter监听器的结果，可以截取服务器监控的截图

8.调优

本次测试结果不理想，服务器因硬件强大，几乎无负载，但应用本身有java出错。并发现接口调用结果未正确影响网页的bug。

后续需要等开发修复、优化之后再次测试

汽车之家

一、测试部分

1、三种注册方式：手机号、邮箱、用户名，每一种方式只能存在唯一的值，这样的设计有没有漏洞

2、订单提交页，提交订单时，会有很多同步校验，只考虑http返回是200ok（即网络通讯、服务、代码都正常）的情况，在什么业务场景下，会提交订单失败

重复提交有可能会失败

提交的订单是个非法数据，服务器处理不了，有可能也会失败

提交太频繁了也会失败（有防刷机制）

二、Linux部分

1、写出你常用的查看日志的方法

（1）tail –f 日志文件名称

（2）vim /var/log/\*\*\* |grep \*\*\*

2、日志是每天一切，log结尾的是当天的日志，如何一次性查找当天所有包含ERROR或EXCEPTION的日志所在行以及前后两行

grep -ir "[ERROR|EXCEPTION]" /home/data/logs/site\*/abc.log

3、如何查看活动端口，如何打印出所有order的工程的进程号+进程名，并杀掉打印出来的进程

（1）查看活动端口

netstat –lnp

（2）打印出所有order的工程的进程号和进程名

ps aux |grep order|grep -v grep|awk '{print $2,$NF}'

（3）杀掉打印出来的进程

kill `ps aux |grep order|grep -v grep|awk '{print $2}'`

ps aux |grep order|grep -v grep|awk '{print "kill "$2}'|sh

4、当配置文件中，一行有多个ab和多个abc，如何使用sed命令，一次性将所有ab替换成ab123，而abc不替换

cat a.txt| sed 's/\<ab\>/ab123/g'

三、python部分

1、列举常用的模块和包

模块：os、sys、time、data、datatime、random、string、math、re、selenium

包：MySQLdb、openpyxl、selenium

2、描述type和isinstance函数的区别

type是用来判断类型的

和isinstance函数一个明显的区别是在判断子类。

type()不会认为子类是一种父类类型。

isinstance()会认为子类是一种父类类型。

3、如何获取一个文件的绝对路径和相对路径？如何将目录下所有文件和子目录下的所有文件中abc替换成大写CBA

（1）获取绝对路径

os.path.abspath(文件名)

（2）获取相对路径

\_\_file\_\_

Sys.argv[0] （更可靠）

补充：\_\_file\_\_和argv[0]差异

在主执行文件中时，两者没什么差异，不过要是在不同的文件下，就不同了，下面示例：

C:\Python27\pythontest>whereutils.py文件内容如下

#coding=utf-8

import os,sys

def show\_where():

print "show\_where: sys.argv[0] is", repr(sys.argv[0])

print "show\_where: \_\_file\_\_ is", repr(\_\_file\_\_)

print "show\_where: cwd is", repr(os.getcwd())

C:\Python27\pythontest>printlog.py 文件内容如下：

import sys, os

import whereutils

print "script: sys.argv[0] is", repr(sys.argv[0])

print "script: \_\_file\_\_ is", repr(\_\_file\_\_)

print "script: cwd is", repr(os.getcwd())

whereutils.show\_where()

所以一般来说，argv[0]要更可靠些

4、用python匹配HTML tag的时候，<.\*>和<.\*?>有什么区别

<.\*>：匹配任意多个不同的字符(贪婪)

<.\*？>：匹配任意多个不同的字符(非贪婪)

5、简单写一个多线程的生产和消费代码

#coding=utf-8

from Queue import Queue

import random

import threading

import time

class Producer(threading.Thread):

def \_\_init\_\_(self,t\_name,queue):

threading.Thread.\_\_init\_\_(self,name=t\_name) #继承threading.Thread基类的构造函数

self.data=queue

def run(self):

for i in range(5):

print '%s:%s is producing %d to the queue!\n' %(time.ctime(),self.getName(),i)

self.data.put(i) #把生成的5个数放到队列中

time.sleep(random.randrange(10)/5) #随机等待一个时间，等待消费者去消费

print '%s:%s finished' %(time.ctime(),self.getName())

class Consumer(threading.Thread):

def \_\_init\_\_(self,t\_name,queue):

threading.Thread.\_\_init\_\_(self,name=t\_name)

self.data=queue #队列赋值给self.data

def run(self):

for i in range(5):

val=self.data.get() #消费者消费数据，就是从队列中get出数据

print '%s:%s is consuming. %d in the queue is consumed!\n' %(time.ctime(),self.getName(),val)

time.sleep(random.randrange(10))

print '%s:%s finished!' %(time.ctime(),self.getName())

def main():

queue=Queue() #产生一个空队列

producer=Producer('Pro',queue) #生产者实例

consumer=Consumer('Con',queue) #消费者实例

producer.start()

consumer.start()

producer.join()

consumer.join()

print 'All,threads terminate'

if \_\_name\_\_=='\_\_main\_\_':

main()

四、SQL部分

1、测试环境一台数据库服务器上，同一端口，两个数据库，product\_db中有一个商品表product，order\_db中有一个表order\_item，通过product\_id关联，因为商品数据被清除，导致订单显示不正常，如何清除所有商品不存在的订单

Delete from order\_item where product\_id in (select id from product)

2、订单表order\_data，订单状态为order\_status，订单金额为should\_pay，如何统计每种状态的订单数和累计订单金额？

Select order\_status,count(should\_pay) from order\_data group by order\_status

五、selenium部分

1、简单描述xpath中，preceding，parent和ancestor的区别，并写出对“活动编号10011065”操作下架所用到的xpath或者css语句（注：页面中有多个下面的数据）

parent::\* 表示当前节点的父节点元素

ancestor::\* 表示当前节点的祖先节点元素

child::\* 表示当前节点的子元素

descendant::\* 表示当前阶段的所有后代元素

self::\* 表示当前节点的自身元素

ancestor-or-self::\* 表示当前节点的及它的祖先节点元素

descendant-or-self::\* 表示当前节点的及它们的后代元素

following-sibling::\* 表示当前节点的后序所有兄弟节点元素

preceding-sibling::\* 表示当前节点的前面所有兄弟节点元素

following::\* 表示当前节点的后序所有元素

preceding::\* 表示当前节点的所有元素

//span[contains(text(),'活动编号')]

2、简单描述webdriver应用中，你所常用的一些设计模式

（1）数据驱动测试框架

（2）关键字测试框架

（3）混合测试框架

数据驱动和关键字测试框架的区别：

数据驱动框架：测试程序是需要人为去写的，数据放到一个配置文件里（excel、txt等等）或者数据库中，这样测试数据和测试程序就分离了

测试的逻辑需要代码来实现

关键字驱动框架：关键字就是框架中程序中的每个方法的名称，定义的是所有的测试步骤，测试步骤的特点是：动作（open\_browser、clear）+操作元素+操作值

关键字都会映射到框架中某一个操作的函数，数据驱动中是没有这种映射的

3、如何对你定位到的页面元素进行标红，如何与隐藏的元素交互

（1）对定位的到的元素进行标红

需要调用js

#coding=utf-8

import os

import time

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

driver=webdriver.Firefox()

driver.get('http://mail.163.com')

assert u"163网易免费邮--中文邮箱第一品牌" in driver.title

#userName=driver.find\_element\_by\_xpath("//\*[@id='idPlaceholder']") #这个不能进行clear

userName=driver.find\_element\_by\_xpath("//\*[@id='idInput']")

js="var q=document.getElementById(\"idInput\");q.style.border=\"1px solid red\";"

driver.execute\_script(js)

js解释：

q=document.getElementById(\"idInput\")

元素q的id 为idInput

q.style.border=\"1px solid red\是元素q的样式，边框为1个像素红色

摘自：http://www.cnblogs.com/fnng/p/3230768.html

（2）用ActionChains模块实现和隐藏元素的交互，见下面的代码

#coding=utf-8

from selenium import webdriver

from selenium.webdriver import ActionChains #ActionChains可以实现双单击、悬浮拖拽

fankui=driver.find\_element\_by\_xpath("//a[@class='ibx-advice-logo']") #定位元素

action\_chains=ActionChains(driver) #生成一个ActionChains对象

action\_chains.move\_to\_element(fankui).perform() #悬浮fankui元素

4、

（1）selectFrame中的relative=up,relative=top,relative=parent等等，是什么意思？他们之间存在怎样的关系？

在当前窗体中选择一个框架(你可以多次调用这个命令用于选择嵌套框架)。

要选择父框架，用”relative=parent”作为定位器；

要选择顶级框架，用”relative=top”。

也可以通过基于0的索引号选择框架；用”index=0”选择第一个框架，或者用”index=2”选择第三个框架

（2）selectWindow中的windowID怎样定位

windowID 是要选择窗体的JavaScript 窗体ID

① title：就是一个window标题栏中的文本。

②name:例:<a target="selenium\_blank60273" href="http://bbs.51testing.com/">软件测试论坛</a>，则selectWindow name='selenium\_blank60273'

③ 当前应用窗体的一个JavaScript变量名：假定该变量包含一个由调用JavaScript window.open()函数所产生的返回值。

④ windowID为null，(或是字符串”null”)，则假定为用户是提交给由浏览器最初实例化的窗体。

5、如何加入插件和Cookie

六、性能测试

1、简述LR常用函数

（1）lr\_log\_message日志函数（不输出行号）

（2）lr\_output\_message日志函数（输出行号）

（3）lr\_error\_message错误日志函数

（4）lr\_save\_string、lr\_save\_int：把c语言的变量转换成LR的变量

（5）lr\_eval\_string：把LR变量转换成C语言变量，但是转换过来的变量是字符型的

（6）lr\_output\_string、lr\_log\_string：输出c语言变量，如果是LR变量的话需要先用lr\_error\_string函数转换一下

（7）web\_reg\_find函数：用于验证在访问页面中是否包含某个关键字

2、如何设计负载？标准是什么？

模拟尽可能多的，可能发生的最极端的情况，还要考虑程序设计的目标（比如说：程序框架需要三年不大变，那么就要预估下3年会达到多少用户，需要团队根据业务发展业务推断去预估）

设计负载的标准：根据需求来定义

3、如何识别性能瓶颈？如何发现web服务器的相关问题？如何发现数据库的相关问题？

数据库:

查看索引创建的是否正确、一条sql语句执行的时间、慢查询

Web：

Web请求处理的时间

4、在哪里设置自动关联选项？哪个函数是用来截取虚拟用户脚本中的动态值

（1）设置自动关联选项：

① 录制脚本

② 录制结束后会弹出Design Studio对话框，选择要关联的项，点击Correlate按钮

这样就可以关联成功，生成相应的代码

也可以点击Design Studio按钮打开关联对话框

选择要关联的然后点击correlate

（2）哪个函数是用来截取虚拟用户脚本中的动态值

web\_reg\_save\_param\_regexp

## python面试感悟

从八月底开始找工作，短短的一星期多一些，面试了9家公司，拿到5份Offer，可能是因为我所面试的公司都是些创业性的公司吧，不过还是感触良多，因为学习Python的时间还很短，没想到还算比较容易的找到了工作，就把这些天的面试经验和大家分享一下，希望为学习Python找工作的小伙伴们提供些许帮助。

笔者感觉面试最主要的两点：1.项目经验。 2.项目经验和招聘职位相符，这是最主要的，其他的都是锦上添花。

自我介绍

这是一道送分题，万年不变的第一个问题。不过有些小伙伴可能没有太在意，其实这个问题已经在面试官心中决定了你的去留意向。自我介绍的主要结构：个人基本信息 + 基本技术构成 + 项目经验（具体项目以及在项目中的负责部分）+ 自我评价，其中的原则就是紧紧围绕招聘岗位的需求做介绍。在此之前要做好准备工作，看看招聘方具体需要什么方向的研发工程师。目前针对Python，拉勾上的招聘多为自动化测试平台的设计与开发、数据的挖掘与清洗。单纯的web开发好像还没有，所以web方向的同学注意，多和运维以及自动化方面靠拢。

二段式询问

在面试的过程当中，在面试官提出问题的时候，往往会就问题本身引申出较深层次的问题。比如：你使用过with语句吗？我的回答是：with语句经常适用于对资源进行访问的场合，确保在访问的过程中不管是否发生异常都会指执行必要的清理操作，比如文件的自动关闭以及线程中锁的自动获取与释放。面试官紧接着问，那你知道为什么with语句能够使文件正确关闭，一下子把我问闷了，只能依稀记得with语句会开辟出一块独立环境来执行文件的访问，类似沙盒机制。面试官对这个答案不置可否，算是勉强通过了。所以知其然更要知其所以然。在平时的学习中，多问一个为什么，面试的时候就不会太被动。

不要给自己挖坑

确保你在回答面试官的过程中，回答中的每个知识点都了然于胸，不然被问住，是很难堪的。我在回答web安全问题时，顺嘴说了SQL注入，面试官说既然提到了SQL注入，那么你讲讲它的原理及解决方法吧！丢脸的是我竟然把XSS跨站注入攻击和SQL注入搞混了，场面也是有点尴尬。所以斟酌你说的每一句话，聪明点的同学还可以引导面试官，让他问出自己想要被问的问题。

必问到Redis，高并发解决办法

面试了好多家公司，必然问道Redis了解多少，高并发的解决办法。笔者回答的都不是很好。

这一年你学习了什么新的技能

这是面试官在考察你是否对于新鲜技术抱有极大热忱。面试我的面试官无一例外都问到了这个问题。他们都希望能找一个不断学习，开括创新的年轻人。多浏览最新的技术资讯，选择一方面自己感兴趣的领域。

你会选择创业公司还是像BAT那样的大公司，为什么？

当然是看招聘方属于哪一个公司啦，不过问这种问题的一般都是创业公司。答案无非是：挑战大，享受挑战；创业公司具有无限成功的可能性，想随公司一起成长；

为什么你要从上一家公司离职？

这也是一个必问问题，找一个比较正当的理由，不要说什么公司零食太多胖了20斤，公司周别附近的外卖都吃腻了，真的别这样说…主要原则就是不要对前公司抱有怨言，BOSS朝令夕改，PM不靠谱什么的，多寻找自身原因：公司发展比较稳定，但我还年轻，希望有更大的挑战和更多的学习机会。像这样就可以。

描述一下你的上一家公司

这个问题问到的几率不太大，不过也还是有三家公司问到过，招聘方主要想从上一家公司的具体经营规模以及主营业务来定位你的水平，知道招聘方的目的就可以从容应答。

技术性问题

非技术性的问题就是以上这么多，作为参考稍加准备，面试的时候就能对答如流。下面讲一下在面试中的技术性问题。个人感觉技术性的问题面试官问的没有特别多，一般考察2-3个，由浅到深。

1简述函数式编程

在函数式编程中，函数是基本单位，变量只是一个名称，而不是一个存储单元。除了匿名函数外，Python还使用fliter(),map(),reduce(),apply()函数来支持函数式编程。

2什么是匿名函数，匿名函数有什么局限性

匿名函数，也就是lambda函数，通常用在函数体比较简单的函数上。匿名函数顾名思义就是函数没有名字，因此不用担心函数名冲突。不过Python对匿名函数的支持有限，只有一些简单的情况下可以使用匿名函数。

3如何捕获异常，常用的异常机制有哪些？

如果我们没有对异常进行任何预防，那么在程序执行的过程中发生异常，就会中断程序，调用python默认的异常处理器，并在终端输出异常信息。

try...except...finally语句:当try语句执行时发生异常，回到try语句层，寻找后面是否有except语句。找到except语句后，会调用这个自定义的异常处理器。except将异常处理完毕后，程序继续往下执行。finally语句表示，无论异常发生与否，finally中的语句都要执行。

assert语句：判断assert后面紧跟的语句是True还是False，如果是True则继续执行print，如果是False则中断程序，调用默认的异常处理器，同时输出assert语句逗号后面的提示信息。

with语句：如果with语句或语句块中发生异常，会调用默认的异常处理器处理，但文件还是会正常关闭。

4copy()与deepcopy()的区别

copy是浅拷贝，只拷贝可变对象的父级元素。 deepcopy是深拷贝，递归拷贝可变对象的所有元素。

5函数装饰器有什么作用（常考）

装饰器本质上是一个Python函数，它可以让其他函数在不需要做任何代码变动的前提下增加额外功能，装饰器的返回值也是一个函数对象。它经常用于有切面需求的场景，比如：插入日志、性能测试、事务处理、缓存、权限校验等场景。有了装饰器，就可以抽离出大量与函数功能本身无关的雷同代码并继续重用。

6简述Python的作用域以及Python搜索变量的顺序

Python作用域简单说就是一个变量的命名空间。代码中变量被赋值的位置，就决定了哪些范围的对象可以访问这个变量，这个范围就是变量的作用域。在Python中，只有模块（module），类（class）以及函数（def、lambda）才会引入新的作用域。Python的变量名解析机制也称为 LEGB 法则：本地作用域（Local）→当前作用域被嵌入的本地作用域（Enclosing locals）→全局/模块作用域（Global）→内置作用域（Built-in）

7新式类和旧式类的区别,如何确保使用的类是新式类

为了统一类(class)和类型(type)，python在2.2版本引进来新式类。在2.1版本中，类和类型是不同的。

为了确保使用的是新式类，有以下方法：

放在类模块代码的最前面 \_\_metaclass\_\_ = type

从内建类object直接或者间接地继承

在python3版本中，默认所有的类都是新式类。

8简述\_\_new\_\_和\_\_init\_\_的区别

创建一个新实例时调用\_\_new\_\_,初始化一个实例时用\_\_init\_\_,这是它们最本质的区别。

new方法会返回所构造的对象，init则不会.

new函数必须以cls作为第一个参数，而init则以self作为其第一个参数.

9Python垃圾回收机制(常考)

Python GC主要使用引用计数（reference counting）来跟踪和回收垃圾。在引用计数的基础上，通过“标记-清除”（mark and sweep）解决容器对象可能产生的循环引用问题，通过“分代回收”（generation collection）以空间换时间的方法提高垃圾回收效率。

• 引用计数

PyObject是每个对象必有的内容，其中ob\_refcnt就是做为引用计数。当一个对象有新的引用时，它的ob\_refcnt就会增加，当引用它的对象被删除，它的ob\_refcnt就会减少.引用计数为0时，该对象生命就结束了。

优点: 简单 实时性

缺点: 维护引用计数消耗资源 循环引用

• 标记-清除机制

基本思路是先按需分配，等到没有空闲内存的时候从寄存器和程序栈上的引用出发，遍历以对象为节点、以引用为边构成的图，把所有可以访问到的对象打上标记，然后清扫一遍内存空间，把所有没标记的对象释放。

• 分代技术

分代回收的整体思想是：将系统中的所有内存块根据其存活时间划分为不同的集合，每个集合就成为一个“代”，垃圾收集频率随着“代”的存活时间的增大而减小，存活时间通常利用经过几次垃圾回收来度量。

Python默认定义了三代对象集合，索引数越大，对象存活时间越长。

10Python中的@property有什么作用?如何实现成员变量的只读属性？

@property装饰器就是负责把一个方法变成属性调用，通常用在属性的get方法和set方法，通过设置@property可以实现实例成员变量的直接访问，又保留了参数的检查。另外通过设置get方法而不定义set方法可以实现成员变量的只读属性。

11\*args and \*\*kwargs

\*args代表位置参数，它会接收任意多个参数并把这些参数作为元组传递给函数。\*\*kwargs代表的关键字参数，允许你使用没有事先定义的参数名，另外，位置参数一定要放在关键字参数的前面。

12有用过with statement吗？它的好处是什么？具体如何实现？

with语句适用于对资源进行访问的场合，确保不管使用过程中是否发生异常都会执行必要的“清理”操作，释放资源，比如文件使用后自动关闭、线程中锁的自动获取和释放等。

13what will be the output of the code below? explain your answer

14在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。

15获取最大公约数、最小公倍数

16获取中位数

17输入一个整数，输出该数二进制表示中1的个数。其中负数用补码表示。

如何拿到半数面试公司Offer——我的Python求职之路

以上就是我面试过程中所被问到的问题，算法题还是比较少的，也只有2家公司要求写算法，数据结构似乎被问到的不是特别多，就问到了一个B+树的结构。数据库问到的是索引相关的优化。稍微有些基础的都能回答上来，但是最好可以深层次的探讨一下。

本文只做抛砖引玉之用，有些见解还不是特别成熟，希望可以为学习Python找工作的伙伴们提供一些帮助，面试过程当中最重要的一点是放平心态，求职过程是双方的，不需要太过紧张，把自己掌握的知识充分表达出来就好。只要你是匹千里马，迟早会被伯乐牵出来遛一遛的。

百度一轮面试题

1. 针对三角形写出测试用例，判断它是不是三角形，是不是等边、等腰三角形

假设a, b, c分别代表三角形三边且都必须大于0。

a <= 0 or b <= o or c <= o 非三角形

（0， 3， 5）（-1， 3， 4）

（3， 0， 5）（3， -1， 5）

（3， 4， 0）（3， 4， -1）

a > 0 and b > 0 and c > 0

1、（a + b <= c ）or （a + c<= b）or（b + c <= a）非三角形

（1, 2， 3）

（1， 3， 2）

（3， 1， 2）

2、（a + b > c ）and （a + c > b）and（b + c > a）

2.1、 a !=c !=b 普通三角形

（2， 3， 4）

2.2、(a = b and a != c) or (a = c and a != b) or ( b = c and b != a) 等腰三角形

（2， 2， 3）

2.3、a = b = c > 0 等边三角形

（5, 5, 5）

2. 如何判断单链表中是否有环，有没有高效的方法

单链表有环的定义是：链表的尾节点指向了链接中间的某个节点。比如下图，如果单链接有环，则在遍历时，在通过节点J之后，会重新回到节点D。

那到底如何判断一个单链表是否有环呢？

方法一：使用快慢指针法，设两个工作指针，一个快一个慢，如果有环的话，它们必然在会在某点相遇。

该方法简单好懂，效率高。

方法二：

也设置两个指针P和q，p总是向前走，但q每次都从头开始走，对于每个节点，看p走的步数是否和q一样。比如p从A走到D，用了4步，而q则用了14步，因而步数不等，出现矛盾，所以存在环。

3. 冒泡排序、快排是什么，请写出冒泡排序的代码

冒泡排序（Bubble Sort），是一种较简单的排序算法。

就像水中的气泡一样，将大的气泡像下层，小气泡像上冒。

对序列中的数两两比较，如果第一个数大于第二个数，就交换它们，这样每一趟下来，就将序列中最大数排到无序序列的最后位置。

快速排序（Quicksort）是对冒泡排序的改进版，基于分治技术的重要排序算法，也是实际工作中比较常用的一种优秀的排序算法，速度快、效率高。

通过一趟排序将待排序记录分割成独立的两部分，其中一部分记录的关键字均比另一部分关键字都小，然后分别对这两部分继续进行递归调用函数本身，进行排序，直到整个序列有序。

int[] a = {2, 6, 3, 1, 8, 10, 9};

for (int i = 0; i < a.length - 1; i++)

{

for (int j = 0; j < a.length -i -1; j++)

{

int tmp = 0;

if (a[j] > a[j + 1])

{

tmp = a[j];

a[j] = a[j + 1];

a[j + 1] = tmp;

}

}

}

def bubbleSort(list1) :

n = len(list1)

for i in xrange(n - 1) :

#控制比较的趟数

for j in xrange(n - i - 1) :

#控制每一趟比较n - i个元素

if list1[j] > list1[j + 1] :

#两两比较，前大于后则交换

tmp = list1[j]

list1[j] = list1[j + 1]

list1[j + 1] = tmp

#!/usr/bin/python

#coding=utf-8

def partition(listNum, start, end) :

#一次划分

i = start

j = end

while i < j :

while i < j and listNum[i] < listNum[j] :

#在右侧寻找第一个比轴值小的数

j -= 1

if i < j :

tmp = listNum[i]

listNum[i] = listNum[j]

listNum[j] = tmp

i += 1

while i < j and listNum[i] < listNum[j] :

#左侧寻找第一个比轴值大的数

i += 1

if i < j :

tmp = listNum[i]

listNum[i] = listNum[j]

listNum[j] = tmp

j -= 1

return i #i为轴值记录的最终位置

def quickSort(listNum, start, end) :

if start < end :

#调用partition函数，进行划分，寻找轴值的位置

keyPartition = partition(listNum, start, end)

#对左侧调用函数本身，递归进行快速排序

quickSort(listNum, start, keyPpartition - 1)

#对右侧调用函数本身，递归进行快速排序

quickSort(listNum, keyPpartition + 1, end)

4. java中，Int和integer的区别

int与integer的区别从大的方面来说就是基本数据类型与其包装类的区别：

int 是基本类型，直接存数值，而integer是对象，用一个引用指向这个对象

(1).Java 中的数据类型分为基本数据类型和复杂数据类型

int 是前者而integer 是后者（也就是一个类）；因此在类进行初始化时int类的变量初始为0.而Integer的变量则初始化为null

int i =1；Integer i= new Integer

1.int是基本的数据类型；

2.Integer是int的封装类；

3.int和Integer都可以表示某一个数值；

4.int和Integer不能够互用，因为他们两种不同的数据类型；

java 自动装箱与拆箱

java语言规范中说道：在许多情况下包装与解包装是由编译器自行完成的（在这种情况下包装成为装箱，解包装称为拆箱）；

其实按照我自己的理解自动装箱就可以简单的理解为将基本数据类型封装为对象类型，来符合java的面向对象；

例：

//声明一个Integer对象

Integer num = 10;

//以上的声明就是用到了自动的装箱：解析为

Integer num = new Integer(10);

自动将基本数据类型转化为对应的封装类型。成为一个对象以后就可以调用对象所声明的所有的方法。

自动拆箱：故名思议就是将对象重新转化为基本数据类型：

//装箱

Integer num = 10;

//拆箱

int b = new Integer(100); 这是自动拆箱

int num1 = num;

自动拆箱有个很典型的用法就是在进行运算的时候：因为对象时不能直接进行运算的，而是要转化为基本数据类型后才能进行加减乘除

Integer num = 10;

//进行计算时隐含的有自动拆箱

System.out.print(num--);

5. 用Linux命令，删除文件中含有abc的那一行

sed -i '/abc/d' sed.txt

6. 用linux命令，统计文件中所有的IP出现次数，显示出现次数最多的前5名

cat ip.txt | grep -o "[0-9][0-9]\*\.[0-9][0-9]\*\.[0-9][0-9]\*\.[0-9][0-9]\*" | awk '{a[$1]++}END{for (i in a) print a[i],i}' | sort -nr | head -5

7. 数据库:将一个表按age排序，取出第二大的age值

select age from t order by age desc limit 1,1;

8. 如何备份数据库，如何恢复，写出具体命令

mysqldump备份命令

mysqldump是mysql的逻辑备份命令，备份成一个可执行的sql文本，也就是说保存的都是可执行的完整的sql语句（如：建库语句、建表语句、插入数据等语句）。

备份现有的所有数据库，并给备份的sql文件命名为mysql\_back.sql

[root@admin training]# mysqldump -uroot -proot --all-databases > mysql\_back.sql

将备份的sql文件恢复回去

[root@admin training]# mysql -uroot -proot [库名或表名]< mysql\_back.sql

注意：恢复的时候，如果原库中存在某个表或者是某个库，恢复的时候就先删除原库或表，然后恢复成备份的数据，即使表中新插入了数据，那也会丢失的，所以在恢复的时候一定要小心，记得备份。但是如果是新建了某个库或者是某个表，这个时候这些库或表则不会被覆盖。

还有一种方法是，先停掉mysql，然后备份物理文件（主从切换，先备份从再备份主）。

还有一种是使用备份工具进行热备份。

9. 说出你知道的数据库join查询方式，分别是什么意思

inner join

内连接，

left join

right join

联接（join）行为是指将两个或多个表中满足条件的数据合并到一个表中，然后使用常用的sql语句在新的表中进行查询。

1、内联接（inner join）：将两个表中存在联接关系、满足联接条件的字段，组成记录集。

默认的join都是inner join。

mysql> select \* from t1 inner join t2 using(id);

或

mysql> select \* from t1 inner join t2 on t1.id=t2.id;

2、外联接：分左联接和右联接

 左联接：left join

left join是以左表为准，将右表中满足联接条件的字段联接形成新的记录集。这里需要注意的是，最后出来的记录会包括左表的全部记录。如联接t1和t2表：

mysql> select \* from t1 left join t2 on t1.id=t2.id;

t1表中id有1,2,3，而t2表中id只有1,2，left join时，由于是以左表为准，所以会显示左表中的全部数据，右表如没有匹配的数据，就会插入null代替。

 右联接：right join

right join是以右表为准，将左表中满足条件的字段联接形成新的记录集。最后出来的记录会包括右表中的全部记录。如，右联接t1和t2：

mysql> select \* from t1 right join t2 on t1.id=t2.id;

Cross join

cross join：交叉连接，得到的结果是两个表的乘积，即笛卡尔积

笛卡尔（Descartes）乘积又叫直积。假设集合A={a,b}，集合B={0,1,2}，则两个集合的笛卡尔积为{(a,0),(a,1),(a,2),(b,0),(b,1), (b,2)}。

select \* from A cross join B on A.name = B.name;

实际上，在 MySQL 中（仅限于 MySQL） CROSS JOIN 与 INNER JOIN 的表现是一样的，在不指定 ON 条件得到的结果都是笛卡尔积，反之取得两个表完全匹配的结果。 INNER JOIN 与 CROSS JOIN 可以省略 INNER 或 CROSS 关键字

Full join

全连接产生的所有记录（双方匹配记录）在表A和表B。如果没有匹配,则对面将包含null。

10. 是否会c和C++

11. 简述性能测试实施过程，分析哪些性能数据，如何导出linux服务器中的cpu，io，内存性能测试数据

概括来说性能测试的实施步骤是以下步骤循环执行的过程：

制定测试计划→测试需求分析→测试案例制定→测试环境配置（包括软硬件环境、测试工具等）→录制测试脚本→设计测试场景→执行测试→获取测试结果→测试结果评估和写测试报告→制定测试计划→测试需求分析→……

分析的数据：

平均负载，程序执行时间，CPU利用率，IO使用情况，

性能测试及分析:

Linux系统出现问题时，我们不仅需要查看系统日志信息，而且还要使用大量的性能监测工

具来判断究竟是哪一部分（内存、CPU、硬盘……）出了问题。在Linux系统中，所有的运行

参数保存在虚拟目录/proc中，换句话说，我们使用的性能监控工具取到的数据值实际上就

是源自于这个目录，当涉及到系统高估时，我们就可以修改/proc目录中的相关参数了，当

然有些是不能乱改的。下面就让我们了解一下这些常用的性能监控工具。工具功能描述uptime----系统平均负载率

dmesg----硬件/系统信息

top-----进程进行状态

iostat--------CPU和磁盘平均使用率

vmstat---------系统运行状态

sar---------实时收集系统使用状态

KDE System Guard---图形监控工具

free---------------内存使用率

traffic-vis------------网络监控（只有SUSE有）

pmap-------------进程内存占用率

strace---------追踪程序运行状态

ulimit---------系统资源使用限制

mpstat-------------多处理器使用率

12. 如何看数据库是否有慢查询

MySQL通过慢查询日志定位那些执行效率较低的SQL 语句，用--log-slow-queries[=file\_name]选项启动时，mysqld 会写一个包含所有执行时间超过long\_query\_time 秒的SQL语句的日志文件，通过查看这个日志文件定位效率较低的SQL 。下面介绍MySQL中如何查询慢的SQL语句

一、MySQL数据库有几个配置选项可以帮助我们及时捕获低效SQL语句

1，slow\_query\_log

这个参数设置为ON，可以捕获执行时间超过一定数值的SQL语句。

2，long\_query\_time

当SQL语句执行时间超过此数值时，就会被记录到日志中，建议设置为1或者更短。

3，slow\_query\_log\_file

记录日志的文件名。

4，log\_queries\_not\_using\_indexes

这个参数设置为ON，可以捕获到所有未使用索引的SQL语句，尽管这个SQL语句有可能执行得挺快。

百度二面

1. 写程序：将一篇英文文章（有标点）倒序输出，不改变单词原来的顺序。

#coding=utf-8

listP = [',', '.', '?', '!', ';', '"', "'", ':']

fp = open('c:\\words.txt', 'r')-

fLines = fp.readlines()

fp.close()

fp = open('c:\\articele1.txt','w')

if len(fLines) == 0 :

print "space"

else :

fLines.reverse()

for line in fLines :

line = line.replace('\n', '')

if line.isspace() :

fp.write('\n\n')

continue

listLine = line.split()

tmpList = []

for i in xrange(len(listLine)) :

if listLine[i][0] in listP or listLine[i][-1] in listP :

#说明单词前后肯定有标点符号

if listLine[i][0] == listLine[i][-1] :

tmpList.append(listLine[i])

elif listLine[i][0] == listLine[i][-2] :

tmpList.append(listLine[i][:-1])

tmpList.append(listLine[i][-1])

elif listLine[i][0] in listP :

tmpList.append(listLine[i][0])

tmpList.append(listLine[i][1:])

elif listLine[i][-1] in listP :

if listLine[i][-2] in listP :

tmpList.append(listLine[i][:-2])

tmpList.append(listLine[i][-2])

else :

tmpList.append(listLine[i][:-1])

tmpList.append(listLine[i][-1])

else :

tmpList.append(listLine[i])

tmpList.reverse()

lineStr = ' '.join(tmpList)

fp.writelines(lineStr + '\n')

fp.close()

2. 写程序：判断一个数是否是2的几次幂，如果是返回true，不是返回false。例如：判断8是不是2的几次幂，结果是“是”，返回true。

#coding=utf-8

'''

移位运算：

<< 左移

运算规则：

按二进制形式把所有的数字向左移动对应的位数，高位移出(舍弃)，低位的空位补零。

在数字没有溢出的前提下，对于正数和负数，左移一位都相当于乘以2的1次方，左移n位就相当于乘以2的n次方。

>> 右移

运算规则：

按二进制形式把所有的数字向右移动对应位数，低位移出(舍弃)，高位的空位补符号位，即正数补零，负数补1。

右移一位相当于除2，右移n位相当于除以2的n次方。

#coding=utf-8

'''

import math

def is2Power(num) :

if num < 2 :

return False

else :

tmp1 = num

tmp2 = 0

while(tmp1 > 1) :

tmp2 = tmp1 % 2

if tmp2 != 0 :

return False

tmp1 /= 2

return True

#把乘除运算换成移位运算

def is2Power1(num) :

if num < 2 :

return False

else :

tmp1 = num

tmp2 = 0

while(tmp1 > 1) :

tmp2 = tmp1 % 2

if tmp2 != 0 :

return False

tmp1 = tmp1 >> 1

return True

'''这种写法看似优化了，但是总觉得有循环在里面不够优化，一次一次的移，不如一次就移到位，

不用加循环在里面，把十进制数转化成2进制数：

因为一个数如果是2的整数幂，那么它的2进制表示里面肯定只有一个1, 并且是最高位，当然不考虑负数。

'''

def is2Power2(num) :

binaryNum = '{0:b}'.format(num)

if binaryNum.count('1') == 1 and binaryNum[0] == '1':

return True

return False

#使用按位与运算符判断

'''

如果一个数是2^n，说明这个二进制里面只有一个1。

a = (10000)ba-1 = (01111)ba&(a-1) = 0。

如果一个数不是2^n，说明它的二进制里含有多个1。

a = (1xxx100)ba-1=(1xxx011)b那么 a&(a-1)就是 (1xxx000)b，而不会为0。

所以可以用这种方法判断一个数是不2^n。

'''

num = int(raw\_input("please input a integer:"))

if is2Power2(num) :

print "True"

else :

print "False"

用例设计：

1.如何测试电梯，写出测试用例

要点：

功能测试

1.电梯能否实现正常上升、下降功能

2.

3.电梯门打开、关闭是否正常

4.电梯按钮是否可用

5.报警设备是否可用

6.与其他电梯之间协作是否良好

7.如何处理突然停电的情况

8.电梯里通风情况如何，灯光如何

9.电梯最终目标是上升到10，当上升到了6楼时，时1楼有人叫梯，电梯是否是先上升到10楼然后再下降到1楼

10.电梯现在在1楼，有人在10楼叫梯，但电梯运行到5楼后，有人在7楼叫梯，电梯是否会现在7楼停

11.电梯下降到8楼时满员，此时有人在5楼叫梯，是否在5楼停

12.关闭电梯门后手机是否有信号

安全性测试

1.关门一瞬间出先障碍物，如何反应

2.出现故障是否有报警措施

3.轿厢停靠位置是否合理

4.出现安全故障时，门是否容易打开

5.同时按关门和开门按钮，电梯是否正常

6.多次点击同一楼层号码，电梯反应情况

7.如何处理突然停电的情况

8.同时按上键和下键，电梯反应情况。

压力测试

1.电梯的最大负载量

2.超负载时，报警设备是否正常提醒

3.在一定时间内不断让电梯上升下降，是否能正常运行

易用性测试

1.电梯按钮设计是否符合大部分人习惯

2.电梯中是否有安全知识说明。

3.轿厢设计是否有考虑残疾人。

稳定性测试

1.电梯在最大负载情况下平稳运行的最长时间

2.电梯加速过程中是否运行平稳

1. 列出你所知道的常见的http状态码及所代表的含义。

2. http请求头通常包括哪些信息？

1. 简述字符串连接符+和join的区别，哪个更高效；

字符串对象是不可改变的，也就是说在python创建一个字符串后，你不能把这个字符中的某一部分改变。任何上面的函数改变了字符串后，都会返回一个新的字符串，原字串并没有变。

所以上面的例子中，任何对字符串的操作包括’+’操作符都将创建一个新的字符串对象，而不是改变原来的对象.因此N个字符串相加必将丢弃中间N-1个结果，而列表不同，列表是可以改变的，因此前面使用list的append，再使用join还原成字符串，只内建了一次，节省了很多资源和时间。

字符串不可改变的现象其实这也是有变通的办法的，可以用S=list(S)这个函数把S变为由单个字符为成员的list，这样的话就可以使用S[3]=’a’的方式改变值，然后再使用S=” “.join(S)还原成字符串。

2. 什么是性能测试，列出常用的性能指标

监控指标

性能测试通常需要监控的指标包括：

1.服务器Linux（包括CPU、Memory、Load、I/O）。

2.数据库：1.Mysql 2.Oracle（缓存命中、索引、单条SQL性能、数据库[/url]线程数、数据池连接数）。

3.中间件：1.Jboss 2. Apache（包括线程数、连接数、日志）。

4.网络： 吞吐量、吞吐率。

5.应用： jvm内存、日志、Full GC频率。

6.监控工具（LoadRunner[/url]）：用户执行情况、场景状态、事务响应时间、TPS等。

7.测试机资源：CPU、Memory、网络、磁盘空间。

3. 列出你所知道的性能测试工具

4. 压力测试、负载测试、稳定测试分别是什么

（压力测试），检查被测系统在恶劣环境下的表现，比如内存不足，CPU高负荷，网速慢，等等。在这种情况下，被测系统往往运行缓慢，能暴露许多在高速运行时不易发生的问题，例如线程不同步。

压力测试是测试系统在非正常的、超负荷的条件下的运行情况 ，用来评估在超越最大负载的情况下系统将如何运行，考验系统在正常的情况下对某种负载强度的承受能力 。压力测试就是一种破坏性的测试，用以判断系统的稳定性和可靠性。一般把压力描述为“CPU使用率达到75%以上，内存使用率达到70%以上”。

5.

Load Testing（负载测试），检查被测系统处理大量用户访问或事务的能力。负载测试是一种性能测试，指数据在超负荷环境中运行，程序是否能够承担。

6. Volume Testing（容量测试），检查被测系统处理大数据量的能力，例如存储或读取一个超长的文件。确定系统可处理同时在线的最大用户数。

7. 强度测试：强度测试是一种性能测试，他在系统资源特别低的情况下软件系统运行情况，目的是找到系统在哪里失效以及如何失效的地方。

declare say varchar(30) default '';

delimiter //

create procedure simple\_pro(in in\_param smallint, out out\_param varchar(20))

begin

select name into out\_param from pro\_test where id=in\_param;

end

//

delimiter ;

#调用改存储过程

mysql> call simple\_pro(2, @ret\_result);

解析：

select name into out\_param

表示将查询的结果name的值赋给输出变量out\_param

调用后，将存储过程的结果赋给实际变量ret\_result

需要注意，返回的结果值的类型，必须和定义的out变量的类型一致，否则会报错。

存储器中定义变量：

declare rowCount int;

set rowCount = 100;

select rowCount \* in\_param into out\_param;

也可以这样写：

declare rowCount int;

set @rowCount = 100;

select @rowCount \* in\_param into out\_param;

此时，结果就是int类型，所以out参数的类型也必须是int类型。

1、查看状态

SHOW {PROCEDURE | FUNCTION} STATUS [LIKE 'pattern'] ;

不加like后的语句，表示查看所有的存储过程。

2、查看定义

SHOW CREATE {PROCEDURE | FUNCTION} sp\_name ;

删除存储过程和函数

DROP {PROCEDURE | FUNCTION} [IF EXISTS] sp\_name ;

delimiter //

create procedure pro\_if(in x int)

begin

if x = 0 then

select 'x = 0' as result;

elseif x > 0 then

select 'x > 0' as result;

else

select 'x < 0' as result;

end if;

end

//

delimiter ;

case ...when

delimiter //

create procedure pro\_case(in x int)

begin

case x

when 1 then select 'success' as result ;

when 0 then select 'Nothing' as result ;

when -1 then select 'faild' as result ;

else select 'try again' as result ;

end case ;

end

//

delimiter ;

触发器

语法结构：

CREATE TRIGGER trigger\_name

trigger\_time trigger\_event ON tab\_name

FOR EACH ROW trigger\_stmt

 trigger\_time 是触发器的动作时间。它可以是before或after

 trigger\_event 指明了激活触发程序的语句的类型。trigger\_event可以是：

insert：通过insert、load data和replace语句

update：通过update语句

delete：通过delete和replace语句

 trigger\_stmt 当触发程序激活时执行的语句。

触发器注意事项：

1、对于具有相同触发程序动作时间和事件的给定表，不能有两个触发程序。例如，对某一个表，不能有两个before update触发程序。

2、在触发器中使用别名OLD.column和NEW.column来记录变化前（更新、删除）的值和变化后（更新、删除）的值。

3、对于insert into .... on duplicate key update 语句的触发顺序。

实例：

创建3张表，如下：

create table tab\_tri1(a1 int, a2 varchar(10));

create table tab\_tri2(b1 varchar(10));

create table tab\_tri3(c1 varchar(10));

insert into tab\_tri1 values(1,'max'),(2,'jack'),(3,'pen');

创建以触发器，如下：

delimiter //

create trigger tri\_test

after update on tab\_tri1

for each row

begin

insert into tab\_tri2 values(old.a2);

insert into tab\_tri3 values(new.a2);

end

//

delimiter ;

这触发器的意思就是，在表tab\_tri1更新后，将旧值插入tab\_tri2表，新值插入tab\_tri3表。

更新表tab\_tri1字段的值：

1、使用show查看

mysql> show triggers \G

2、通过information\_schema库查看

mysql> select \* from information\_schema.triggers where trigger\_name="tri\_test" \G;

删除触发器

mysql> drop trigger [schema\_name.]trigger\_name ;

视图

视图是一种虚拟存在的表，对于使用视图的用户来说基本上是透明的。在数据库中并不占用存储空间。

优势：

1、简单

2、安全

3、数据独立

4、可以从好几个表中只提取需要的数据，并且会实时更新

创建视图需要有create view权限，并且对于查询涉及的列有select权限。如果使用create or replace 或alter，还需要该视图的drop权限。

语法结构：

CREATE [ OR REPLACE ]

VIEW view\_name[ (column\_list) ]

AS select\_statement

[WITH [CASCADED | LOCAL] CHECK option ]

实例：

delimiter //

create view v\_student\_lesson as

select stu\_info.stuid, stu\_info.name,stu\_lesson.lesson from

stu\_info,stu\_lesson where stu\_info.stuid=stu\_lesson.stuid;

//

delimiter ;

创建的视图表，只是一个虚拟表，并没有真正的物理存储，并且也不支持索引。

更新、删除没有聚合函数的视图时（也就是具有行一一对应的情况的视图），原表的数据也会更新、删除。同时更新、删除原表时，视图的数据也会跟着更新、删除。

设计索引的原则

 搜索的索引列，不一定是索要选择的列

 使用唯一索引（主键是查询最快的）

 使用短索引（越短的索引，存储空间越小，内存中查询的越快）

 利用最左前缀（可以只给前几个字符创建一个索引，必须是前几个字符就能区别）

 不要过度索引（索引个数不要太多，一般5个之内）

 注意InnoaDB表的保存顺序（根据主键在物理上存储分布的）

下面我们就来介绍一下MySQL几种常见的索引吧。

索引分单列索引和组合索引。单列索引，即一个索引只包含单个列，一个表可以有多个单列索引，但这不是组合索引。所谓组合索引，就是一个索引包含多个列。

1、普通索引（index）

它是最基本的索引，它没有任何限制。

2、唯一索引（unique）

字段可以指定唯一索引，但唯一索引值可以为null，而主键不能出现null，这值会特殊处理。

3、主键索引（primary key）

主键索引是一种特殊的唯一索引，不允许为空，并且一般在建表时就创建主键索引。当然也可以使用alter进行修改表字段，指定主键。

一个表只能有一个主键存在。

需要注意，索引名字不能重复。

创建索引有三种方法：

1、创建索引

语法结构：

create index index\_name on table\_name (字段名[index\_length]);

如果是BLOB和TEXT类型，必须指定 length，下同。

实例：

create index idx\_age on idx\_test(age);

create index idx\_name on idx\_test(name);

需要注意，使用create创建索引时，不可以将需要创建的多个索引写到一个create语句中。

2、修改表结构指定索引

alter table table\_name add index/ unique index /key index\_name(字段名[index\_length]);

可以将修改多个字段的索引写到一个alter语句中，如：

alter table idx\_test add index/unique index/key idx\_age(age),add add index/unique/key idx\_name(name);

3、创建表时直接指定索引

例如：

create table idx\_test(

id int auto\_increment primary key,

name varchar(20) not null comment '员工姓名',

age tinyint(3) default 0 comment '员工年龄',

key/index/unique idx\_name(name),

[key/index/unique idx\_age(age)]

)engine=innodb character set=utf8 comment '员工信息表';

联合索引

联合索引，就是一次指定多个字段作为索引，但是这个联合索引字段的位置是有先后顺序的，先写哪个字段，后写哪个字段，字段位置不同，存储时就会不一样。

语法结构：

alter table table\_name add index/ unique index /key idx\_name1\_name2(column\_1, column\_2);

删除索引

删除索引在alter语句中使用drop子句。

语法结构：

alter table table\_name drop index index\_name1, drop index index\_name2.....;

实例：

alter table idx\_test drop index idx\_name,drop index idx\_age;

慢查询

分析MySQL语句查询性能的方法除了使用explain输出执行计划，还可以让Mysql记录下查询超过指定时间的语句，我们将超过指定时间的sql查询称为“慢查询”。

查看慢查询的设置

mysql> show variables like "%slow%" ;

从查询结果看出，此时慢查询是关闭的，开启慢查询

mysql> set global slow\_query\_log=on;

最后一行，就是慢查询日志存储的地方。

设置慢查询的时间

mysql> show variables like "%long%";

默认设置的是10秒，也就是说查询时间超过10秒的查询就会被记录到慢查询日志中。

设置慢查询的阈值

mysql> set long\_query\_time=0;

当我们把慢查询的阈值改成0，表示所有的查询都会记录到慢查询日志。

Mysqldump（逻辑备份）

mysqldump是mysql的逻辑备份命令，备份成一个可执行的sql文本，可以对多表进行备份。

备份现所有的数据库

[root@admin training]# mysqldump -uroot -proot --all-databases > mysql\_back.sql

备份单个test库

[root@admin training]# mysqldump -uroot -proot test > mysql\_back.sql

这样备份的数据中，没有重新建库的语句，也就是说再下次还原数据时，test库时必须存在的，否则就会报错，因为只备份库下面的数据了。为解决这个问题，我们可以加上参数 --database（-B），目的是在备份的数据文件中，增加一句，如果数据库不存在，就创建该数据库。并且还可以同时备份好几个数据库或者是某个库中的某一个表，同时在还原时，也不需要指定哪个库了。

[root@admin training]# mysqldump -uroot -proot --database test > mysql\_back.sql

恢复备份的sql文件

方法一，直接在Linux命令行下恢复

[root@admin training]# mysql -uroot -proot [库名或表名]< mysql\_back.sql

方法二，登录mysql，使用source命令恢复

mysql> source 备份的mysql数据文件名

Select .....into outfile（逻辑备份）

select..... into outfile语句把表数据导出到一个文本文件中，并用load data infile..... 语句恢复数据。但是这种方法只能导出或导入数据的内容，不包括表的结构，如果表的结构文件损坏，则必须先恢复原来的表结构，并且只能针对单表操作。

语法结构：

select [列名] from table\_name [where 语句]

into outfile '目标文件路劲及文件名' [option]

使用select..... into outfile语句来导出test库下的gloryRoad表记录。其中，字段之间用逗号“，”隔开。sql代码如下：

select \* from gloryRoad into outfile '/ROOT/data/mysql/gloryRoad \_bak.txt' fields terminated by ',';

上面的语句表示把表gloryRoad中的全部数据导入/ROOT/data/mysql/gloryRoad \_bak.txt文件中，指定字段间的分隔符为逗号“，”。当然还可以加入where条件，表示导出具体某些字段的数据，或者加入

lines

starting by' \> ' #指定每条记录以“ > ”开头

terminated by ' \r\n ' #保证每行记录占一行，因为Windows下“\n\r”才是回车换行，如果不加这个参数默认情况是“\n”

如：

mysql> select \* from gloryRoad into outfile '/ROOT/data/mysql/gloryRoad.txt' fields terminated by ' ' lines terminated by '\r\n';

当然我还可以在导出数据时，指定导出文件的字符集，如：

mysql> select \* from gloryRoad into outfile '/ROOT/data/mysql/gloryRoad\_bak3.txt' character set gbk fields terminated by ',';

load data infile .....语句导入数据

语法结构：

load data infile '目标文件' into table table\_name [字符集][option][指定导入的字段]

mysql> load data infile '/ROOT/data/mysql/test.txt' into table gloryRoad character set gbk fields terminated by ',';

主从复制的原理

 是一个异步复制过程

 由三个线程完成：其中两个线程（sql线程和IO线程）在slave（从）端，另一个线程（IO线程）在master（主）端。

 必须打开master端的Binary Log（--log-bin）日志

工作步骤：

1、master上只要有数据产生变更（insert、update、alter等，select是不会记录的），就会记录到Binary Log日志中

2、然后通过主从之间的IO线程，将二进制日志文件拉倒slave（从）上，并写入Relay log日志文件中。

3、然后slave的sql线程会将Relay log在slave上执行一遍，从而实现了主从复制。

14、grep抓取文件内容中特定字符

语法：grep <-option> Keywords filename/ filedirectory

filename/ filedirectory：文件名或文件所在的绝对路径。

要匹配的关键字Keywords可以加引号，也可以不加引号。

参数：

 -E ： 扩展的grep，支持复杂正则

 -w : 加上单词的界定，也就是说只有完全匹配给定的单词的行才能被抓取出来

 -F : 完全关闭正则

 -v : 显示不匹配的项

 -i : 不区分大小写

 -o : 只显示匹配内容（模式匹配，给定什么显示什么）

 -r : 递归参数（会递归遍历目录下的所有文件，包括二级目录、三级目录等下所有的文件）

 -A : 搜出关键字以后，还可以向下打印指定的行数，后面接要打印的行数

 -B ：搜出关键字以后，还可以向上打印指定的行数，后面接要打印的行数

#在a.txt文件中搜包含name的行打印出来

[root@admin t2]# grep name a.txt

name is aa

#抽取/root/install.log文件中匹配man关键字的内容

[root@admin /]# cat /root/install.log | grep -w man

Installing man-1.6f-32.el6.x86\_64

Installing man-pages-overrides-6.6.3-2.el6.noarch

Installing man-pages-3.22-20.el6.noarch

#当只知道某个关键字，比如man，但不知道在哪个文件中定义的时候，可以使用grep递归的搜索

[root@admin /]# grep -r man \* <-- \* 代表的是通配当前面目录下的所有文件，也可以在某一个文件中搜索，如a.txt

#使用-o进行模式匹配抽取

[root@admin tt]# grep -o 'this is' 11.txt

this is

this is

this is

#匹配到关键name之以后，再向下打印n = 3行

[root@admin t2]# cat a.txt | grep -A 3 name

name

cfwesdf

12edf

342sdfs

#匹配到关键name之以后，再向上打印n = 3行

[root@admin t2]# cat a.txt | grep -B 3 name

adfasd this is last one %

234123 ^

adsftgrt

name

15、find查找文件命令

find是针对文件的名字中的关键字、属性、创建时间、创建的用户等特性去定位某个文件的。

 按照文件名字查找

find Directory(查找的目录路径) -name keyword(关键字，必须用引号引起来)

 按照文件的类型查找

find Directory(查找的目录路径) -type f/d/l （type ：类型，f/d : 文件/目录/软链接）

 按照创建时间查找

find Directory(查找的目录路径) -type f -ctime -1

find Directory(查找的目录路径) -type d -cmin +30 -name keyword(关键字)

 按照修改时间查找

find Directory(查找的目录路径) -mtime/mmin +3/-2 （time表示天，+3表示3天/分钟以前的/-2表示2天/分钟以内）(注：不算当天、当前分钟)

 通过参数exec执行find的后续操作

find Directory(查找的目录路径) -mtime +30 -exec （需要执行的命令，如：rm -rf） {} \;

此处的花括号“{}”代表find出的文件，“ \;”是一个固定写法，代表转义的分号

#通过文件名匹配找到以a开头的所有文件，（注：find命令默认会递归）

[root@admin home]# find t1/ -name "a\*"

t1/a.txt

#某个路劲下的列出文件或目录（注：这些都默认都是会递归的）

[root@admin home]# find t1/ -type f

t1/a.txt

t1/training.tar.gz

t1/dd/b.txt

t1/training.tar.gz

[root@admin home]# find training/ -type d

training/

training/test

training/test/testtt

#查找一天以前的以aa开头的文件或目录

[root@admin home]# find t1/ -type f/d -mtime +1 -name "aa\*"

#删除30天前的[文件/目录]（首先得找到30天前的文件）

[root@admin home]# find t1/ -mtime +30 [-type f/d] -exec rm -rf {} \;

#或者使用管道命令实现

[root@admin home]# find t1/ -mtime +1 -type f | ls -al

1、4个等级的得分，如何测试？（例如得分前30%是A，30%-50%是B，50%-80%是C，80%-100%是D）

2、一共n个台阶，一次可以上一个台阶或两个台阶，共有多少种上台阶数量到达顶部的可能?

def fn(n):

if n == 1:

return 1

elif n == 2:

return 2

elif n > 2:

return fn(n-2)+fn(n-1)

3、随便写个排序算法

def insert\_sort(lists):

# 插入排序

count = len(lists)

for i in range(1, count):

key = lists[i]

for j in range(i)[::-1]:

if lists[j]>key:

lists[j+1],lists[j]=lists[j],key

return lists

lists = [3,5,6,9,8,2,11,3]

print insert\_sort(lists)

4、列表去重

L1 = ['a',1,2,[2,3,4],'a',[2,3],(1,2),(1,2),{'a':1}]

L2 = []

[L2.append(i) for i in L1 if i not in L2]

5、人员表:id,name；部门：id,name；人员信息表：id,部门id，人员id

一个sql同时查询出和人员信息表中id关联的人员name，和部门id关联的部门name

SELECT stu.stu\_name, org.org\_name FROM stu INNER JOIN stu\_org ON stu.id=stu\_org.stu\_id

INNER JOIN org ON stu\_org.org\_id=org.id;

6、被测系统的架构

7、讲讲你写的接口测试框架，有什么功能，如何实现的，遇到过哪些坑？

8、公司测试环境详细讲解及docker应用

9、接口之间互相依赖时如何测试，如何保证下一个模块测试充分？例如A、B接口互相依赖，A接口会有4种情况，B接口需要针对这四种情况分别做测试，但这两个接口是不同的人负责测试，如何保证B接口的测试负责人能将B测试充分。

10、如何梳理规范测试流程，有哪些表现。

11、是否会漏测，你是如何保证不漏测的？如何保证别人不漏测？线上出现过什么大问题吗？

<http://3641932.blog.51cto.com/3631932/1912486>

12、针对算法模型的测试，如何保证算法的正确性？如何保证自己写的测试程序的正确性，万一模型本身是错的呢？

13、举例说明你在测试过程中如何测试接口的，发现过什么样的bug？

14、接口自动化测试的覆盖率是多少，如何统计测试覆盖率？

15、平时做过哪些异常测试？举例说明。

16、接口测试阶段会发现接口所有的bug吗？

# 2 编程算法

## 2.1字符串

# 给你一串字符串，要求将字符串中任一长度为3的子串中，

# 第一个字符与第三个字符相同的子串移除，编程实现

# 如：abad==>d absdsag==>g

*s='rtyuyraswedde'*

*i=1*

*while i<len(s)-1 and len(s)>=2:*

*if s[0]==s[2]:*

*s=s[3:]*

*continue*

*elif s[i-1]==s[i+1]:*

*s=s[:i-1]+s[i+2:]*

*i=i-2*

*else:*

*i+=1*

*print s*

将给定的字符串按单词反转，如：how are you🡺you are how

*>>> l=word.split(' ')*

*>>> l.reverse()*

*>>> ‘ ’.join(l)*

*>>> for i in reversed(word.split(' ')):*

*... print i,*

*...*

*you are how*

## 2.2列表

#移除列表中的偶数项，在本列表中操作，[2,3,6,7,4,1,9]

*l=[2,3,6,7,4,1,9]*

*for i in range(1,len(l)/2+1):*

*l.remove(l[i])*

*print l #[2, 6, 4, 9]*

1：Python文件操作的面试题

1. 如何用Python删除一个文件？

使用os.remove(filename)或者os.unlink(filename);

2. Python如何copy一个文件？

shutil模块有一个copyfile函数可以实现文件拷贝

2：Python里面如何生成随机数？

标准库random实现了一个随机数生成器，实例代码如下：

import random

random.random()

它会返回一个随机的0和1之间的浮点数

3：如何用Python来发送邮件？

可以使用smtplib标准库。

以下代码可以在支持SMTP监听器的服务器上执行。

import sys, smtplib

fromaddr = raw\_input(”From: “)

toaddrs = raw\_input(”To: “).split(’,')

print “Enter message, end with ^D:”

msg = ”

while 1:

line = sys.stdin.readline()

if not line:

break

msg = msg + line

# 发送邮件部分

server = smtplib.SMTP(’localhost’)

server.sendmail(fromaddr, toaddrs, msg)

server.quit()

4：Python里面如何拷贝一个对象？

一般来说可以使用copy.copy()方法或者copy.deepcopy()方法，几乎所有的对象都可以被拷贝

一些对象可以更容易的拷贝，Dictionaries有一个copy方法：

newdict = olddict.copy()

5：有两个序列a,b，大小都为n,序列元素的值任意整形数，无序；要求：通过交换a,b中的元素，使[序列a元素的和]与[序列b元素的和]之间的差最小。

将给定的字符串按单词反转

“How are you, I'm Lily" 输出

程序题：

1. 写程序：将一篇英文文章（有标点）倒序输出，不改变单词原来的顺序。

2. #coding=utf-8

3. listP = [',', '.', '?', '!', ';', '"', "'", ':']

4. fp = open('c:\\words.txt', 'r')-

5. fLines = fp.readlines()

6. fp.close()

7. fp = open('c:\\articele1.txt','w')

8. if len(fLines) == 0 :

9. print "space"

10. else :

11. fLines.reverse()

12. for line in fLines :

13. line = line.replace('\n', '')

14. if line.isspace() :

15. fp.write('\n\n')

16. continue

17. listLine = line.split()

18. tmpList = []

19. for i in xrange(len(listLine)) :

20. if listLine[i][0] in listP or listLine[i][-1] in listP :

21. #说明单词前后肯定有标点符号

22. if listLine[i][0] == listLine[i][-1] :

23. tmpList.append(listLine[i])

24. elif listLine[i][0] == listLine[i][-2] :

25. tmpList.append(listLine[i][:-1])

26. tmpList.append(listLine[i][-1])

27. elif listLine[i][0] in listP :

28. tmpList.append(listLine[i][0])

29. tmpList.append(listLine[i][1:])

30. elif listLine[i][-1] in listP :

31. if listLine[i][-2] in listP :

32. tmpList.append(listLine[i][:-2])

33. tmpList.append(listLine[i][-2])

34. else :

35. tmpList.append(listLine[i][:-1])

36. tmpList.append(listLine[i][-1])

37. else :

38. tmpList.append(listLine[i])

39. tmpList.reverse()

40. lineStr = ' '.join(tmpList)

41. fp.writelines(lineStr + '\n')

42.

43. fp.close()

44. 写程序：判断一个数是否是2的几次幂，如果是返回true，不是返回false。例如：判断8是不是2的几次幂，结果是“是”，返回true。

#coding=utf-8

'''

移位运算：

<< 左移

运算规则：

按二进制形式把所有的数字向左移动对应的位数，高位移出(舍弃)，低位的空位补零。

在数字没有溢出的前提下，对于正数和负数，左移一位都相当于乘以2的1次方，左移n位就相当于乘以2的n次方。

>> 右移

运算规则：

按二进制形式把所有的数字向右移动对应位数，低位移出(舍弃)，高位的空位补符号位，即正数补零，负数补1。

右移一位相当于除2，右移n位相当于除以2的n次方。

#coding=utf-8

'''

import math

def is2Power(num) :

if num < 2 :

return False

else :

tmp1 = num

tmp2 = 0

while(tmp1 > 1) :

tmp2 = tmp1 % 2

if tmp2 != 0 :

return False

tmp1 /= 2

return True

#把乘除运算换成移位运算

def is2Power1(num) :

if num < 2 :

return False

else :

tmp1 = num

tmp2 = 0

while(tmp1 > 1) :

tmp2 = tmp1 % 2

if tmp2 != 0 :

return False

tmp1 = tmp1 >> 1

return True

'''这种写法看似优化了，但是总觉得有循环在里面不够优化，一次一次的移，不如一次就移到位，

不用加循环在里面，把十进制数转化成2进制数：

因为一个数如果是2的整数幂，那么它的2进制表示里面肯定只有一个1, 并且是最高位，当然不考虑负数。

'''

def is2Power2(num) :

binaryNum = '{0:b}'.format(num)

if binaryNum.count('1') == 1 and binaryNum[0] == '1':

return True

return False

#使用按位与运算符判断

'''

如果一个数是2^n，说明这个二进制里面只有一个1。

a = (10000)ba-1 = (01111)ba&(a-1) = 0。

如果一个数不是2^n，说明它的二进制里含有多个1。

a = (1xxx100)ba-1=(1xxx011)b那么 a&(a-1)就是 (1xxx000)b，而不会为0。

所以可以用这种方法判断一个数是不2^n。

'''

num = int(raw\_input("please input a integer:"))

if is2Power2(num) :

print "True"

else :

print "False"

1：什么是lambda函数？

Python允许你定义一种单行的小函数。定义lambda函数的形式如下：labmda 参数：表达式lambda函数默认返回表达式的值。你也可以将其赋值给一个变量。lambda函数可以接受任意个参数，包括可选参数，但是表达式只有一个：

>>> g = lambda x, y: x\*y

>>> g(3,4)

12

>>> g = lambda x, y=0, z=0: x+y+z

>>> g(1)

1

>>> g(3, 4, 7)

14

也能够直接使用lambda函数，不把它赋值给变量：

>>> (lambda x,y=0,z=0:x+y+z)(3,5,6)

14

如果你的函数非常简单，只有一个表达式，不包含命令，可以考虑lambda函数。否则，你还是定义函数才对，毕竟函数没有这么多限制。

2：Python是如何进行类型转换的？

Python提供了将变量或值从一种类型转换成另一种类型的内置函数。int函数能够将符合数学格式数字型字符串转换成整数。否则，返回错误信息。

>>> int(”34″)

34

>>> int(”1234ab”) #不能转换成整数

ValueError: invalid literal for int(): 1234ab

函数int也能够把浮点数转换成整数，但浮点数的小数部分被截去。

>>> int(34.1234)

34

>>> int(-2.46)

-2

函数°oat将整数和字符串转换成浮点数：

>>> float(”12″)

12.0

>>> float(”1.111111″)

1.111111

函数str将数字转换成字符：

>>> str(98)

‘98′

>>> str(”76.765″)

‘76.765′

整数1和浮点数1.0在python中是不同的。虽然它们的值相等的，但却属于不同的类型。这两个数在计算机的存储形式也是不一样。

3：Python如何定义一个函数

函数的定义形式如

下：

def <name>(arg1, arg2,… argN):

<statements>

函数的名字也必须以字母开头，可以包括下划线“ ”,但不能把Python的

关键字定义成函数的名字。函数内的语句数量是任意的，每个语句至少有

一个空格的缩进，以表示此语句属于这个函数的。缩进结束的地方，函数

自然结束。

下面定义了一个两个数相加的函数：

>>> def add(p1, p2):

print p1, “+”, p2, “=”, p1+p2

>>> add(1, 2)

1 + 2 = 3

函数的目的是把一些复杂的操作隐藏，来简化程序的结构，使其容易

阅读。函数在调用前，必须先定义。也可以在一个函数内部定义函数，内

部函数只有在外部函数调用时才能够被执行。程序调用函数时，转到函数

内部执行函数内部的语句，函数执行完毕后，返回到它离开程序的地方，

执行程序的下一条语句。

4：如何反序的迭代一个序列？how do I iterate over a sequence in reverse order

如果是一个list, 最快的解决方案是：

list.reverse()

try:

for x in list:

“do something with x”

finally:

list.reverse()

如果不是list, 最通用但是稍慢的解决方案是：

for i in range(len(sequence)-1, -1, -1):

x = sequence[i]

<do something with x>

5：Python里面如何实现tuple和list的转换？

函数tuple(seq)可以把所有可迭代的(iterable)序列转换成一个tuple, 元素不变，排序也不变。

例如，tuple([1,2,3])返回(1,2,3), tuple(’abc’)返回(’a’.’b',’c').如果参数已经是一个tuple的话，函数不做任何拷贝而直接返回原来的对象，所以在不确定对象是不是tuple的时候来调用tuple()函数也不是很耗费的。

函数list(seq)可以把所有的序列和可迭代的对象转换成一个list,元素不变，排序也不变。

例如 list([1,2,3])返回(1,2,3), list(’abc’)返回['a', 'b', 'c']。如果参数是一个list, 她会像set[:]一样做一个拷贝

6：Python面试题：请写出一段Python代码实现删除一个list里面的重复元素

可以先把list重新排序，然后从list的最后开始扫描，代码如下：

if List:

List.sort()

last = List[-1]

for i in range(len(List)-2, -1, -1):

if last==List[i]: del List[i]

else: last=List[i]