

## 九月十月百度人搜，阿里巴巴，腾讯华为小米搜狗笔试面试八十题

### 引言

自发表上一篇文章至今（事实上，上篇文章更新了近 3 个月之久），blog 已经停了 3 个多月，而在那之前，自开博以来的 21 个月每月都不曾断过。正如上一篇文章支持向量机通俗导论（理解 SVM 的三层境界）末尾所述：“额，blog 许久未有更新了，因为最近实在忙，无暇顾及 blog。”与此同时，工作之余，也一直在闲心研究数据挖掘：“神经网络将可能作为 Top 10 Algorithms in Data Mining 之番外篇第 1 篇，同时，k-最近邻法(k-nearest neighbor, kNN)算法谈到 kd 树将可能作为本系列第三篇。这是此系列接下来要写的两个算法，刚好项目中也要用到 KD 树”。

但很显然，若要等到下一篇数据挖掘系列的文章时（更新：下一篇 kd 树目前已经完成：<http://t.cn/zjLQ8Ky>），说不定要到年底去了，而最近的这段时间，9 月、10 月，正是各种校招/笔试/面试火热进行的时节，自己则希望能帮助到这些找工作的朋友，故此，怎能无动于衷，于是，3 个多月后，blog 今天更新了。

再者，虽然 blog 自 10 年 10 月开通至 11 年 10 月，一年的时间内整理了 300 多道面试题（这 300 道题全部集锦在此文中第一部分：[http://blog.csdn.net/v\\_july\\_v/article/details/6543438](http://blog.csdn.net/v_july_v/article/details/6543438)）。但毕竟那些题已经是前年或去年的了，笔试面试题虽然每年类型变化不大，但毕竟它年年推陈出新，存着就有其合理性。

OK，以下是整理自 8 月下旬至 10 月份内的各大公司的笔试面试三十题（注：所有题目基本上全部为软件开发方向，题目来源：网络收集），相信一定能给正在参加各种校招的诸多朋友多少帮助，学习参考或借鉴（如果你手头上有好的笔试/面试题，欢迎通过微博私信：<http://weibo.com/julyweibo>，或邮箱：[zhoulei0907@yahoo.cn](mailto:zhoulei0907@yahoo.cn) 发给我，或者干脆直接评论在本文下；同时，若你对以下任何一题有任何看法.想法.思路或建议，欢迎留言评论，大家一起讨论，共同享受思考的乐趣，谢谢）。

## 九月十月百度人搜，阿里巴巴，腾讯华为小米搜狗笔试面试八十题

「我正在一点一点做.整理下面的笔试面试题，欢迎读者朋友们跟我一起做，你可以把你的答案或代码直接评论在本文之下，也可以通过私信或邮件发给我，感谢诸位。同时，以下所有任何题目所给的点评里的答案，尤其是所给的外部链接若有任何问题，欢迎在本文评论下留言指正，谢谢。答题除了让你感受到思考的乐趣以外，还有奖哦，请君自看。July、二零一二年十月十一日」

### 1. 9 月 11 日， 京东：

谈谈你对面向对象编程的认识

2. 8月20日，金山面试，题目如下：

数据库1中存放着a类数据，数据库2中存放着以天为单位划分的表30张（比如table\_20110909,table\_20110910,table\_20110911），总共是一个月的数据。表1中的a类数据中有一个字段userid来唯一判别用户身份，表2中的30张表（每张表结构相同）也有一个字段userid来唯一识别用户身份。如何判定a类数据库的多少用户在数据库2中出现过？

来源：

<http://topic.csdn.net/u/20120820/23/C6B16CCF-EE15-47C0-9B15-77497291F2B9.html>。

3. 百度实习笔试题（2012.5.6）

1、一个单词单词字母交换，可得另一个单词，如army->mary，成为兄弟单词。提供一个单词，在字典中找到它的兄弟。描述数据结构和查询过程。评点：同去年9月份的一道题，见此文第3题：[http://blog.csdn.net/v\\_july\\_v/article/details/6803368](http://blog.csdn.net/v_july_v/article/details/6803368)。

2、线程和进程区别和联系。什么是“线程安全”

3、C和C++怎样分配和释放内存，区别是什么

4、算法题1

一个url指向的页面里面有另一个url,最终有一个url指向之前出现过的url或空，这两种情形都定义为null。这样构成一个单链表。给两条这样单链表，判断里面是否存在同样的url。url以亿级计，资源不足以hash。

5、算法题2

数组al[0,mid-1]和al[mid,num-1]，都分别有序。将其merge成有序数组al[0,num-1]，要求空间复杂度O(1)

6、系统设计题

百度搜索框的suggestion，比如输入“北京”，搜索框下面会以北京为前缀，展示“北京爱情故事”、“北京公交”、“北京医院”等等搜索词，输入“结构之”，会提示“结构之法”，“结构之法 算法之道”等搜索词。

请问，如何设计此系统，使得空间和时间复杂度尽量低。



新闻 网页 贴吧 知道 MP3 图片 视频 地图

结构之|

结构之法

结构之法 算法之道

百度一下

评点：老题，直接上 **Trie 树**「Trie 树的介绍见：从 Trie 树（字典树）谈到后缀树」+**TOP K**「hashmap+堆，hashmap+堆 统计出如 10 个近似的热词，也就是说，只存与关键词近似的比如 10 个热词」？ or **Double-array trie tree**？同时，**StackOverflow** 上也有两个讨论帖子：<http://stackoverflow.com/questions/2901831/algorithm-for-autocomplete>，<http://stackoverflow.com/questions/1783652/what-is-the-best-autocomplete-suggest-algorithm-datastructure-c-c>。此外，这里有一篇关于“拼写错误检查”问题的介绍，或许对你有所启示：

<http://blog.afterthedeathline.com/2010/01/29/how-i-trie-to-make-spelling-suggestions/>。

#### 4. 人搜笔试

1. 快排每次以第一个作为主元，问时间复杂度是多少？（ $O(N \log N)$ ）

2.  $T(N) = N + T(N/2) + T(2N)$ ，问  $T(N)$  的时间复杂度是多少？ 点评： $O(N \log N)$  or  $O(N)$ ？

3. 从  $(0,1)$  中平均随机出几次才能使得和超过 1？（ $e$ ）

4. 编程题：

一棵树的节点定义格式如下：

```
struct Node{
    Node* parent;
    Node* firstChild; // 孩子节点
    Node* sibling; // 兄弟节点
}
```

要求非递归遍历该树。

思路：采用队列存储，来遍历节点。

5. 算法题：

有  $N$  个节点，每两个节点相邻，每个节点只与 2 个节点相邻，因此， $N$  个顶点有  $N-1$  条边。每一条边上都有权值  $w_i$ ，定义节点  $i$  到节点  $i+1$  的边为  $w_i$ 。

求：不相邻的权值和最大的边的集合。

#### 5. 人搜面试，所投职位：搜索研发工程师：面试题回忆

1、删除字符串开始及末尾的空白符，并且把数组中间的多个空格（如果有）符转化为 1 个。

2、求数组（元素可为正数、负数、0）的最大子序列和。

3、链表相邻元素翻转，如  $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g$ ，翻转后变为：

$b \rightarrow a \rightarrow d \rightarrow c \rightarrow f \rightarrow e \rightarrow g$

4、链表克隆。链表的结构为：

```
typedef struct list {
    int data; //数据字段
    list *middle; //指向链表中某任意位置元素(可指向自己)的指针
    list *next; //指向链表下一元素
} list;
```

5、100 万条数据的数据库查询速度优化问题，解决关键点是：根据主表元素特点，把主表拆分并新建副表，并且利用存储过程保证主副表的数据一致性。（不用写代码）

6、求正整数  $n$  所有可能的和式的组合（如： $4=1+1+1+1$ 、 $1+1+2$ 、 $1+3$ 、 $2+1+1$ 、 $2+2$ ）。点评：这里有一参考答案：

<http://blog.csdn.net/wumuzi520/article/details/8046350>。

7、求旋转数组的最小元素（把一个数组最开始的若干个元素搬到数组的末尾，我们称之为数组的旋转。输入一个排好序的数组的一个旋转，输出旋转数组的最小元素。例如数组{3, 4, 5, 1, 2}为{1, 2, 3, 4, 5}的一个旋转，该数组的最小值为 1）

8、找出两个单链表里交叉的第一个元素

9、字符串移动（字符串为\*号和 26 个字母的任意组合，把\*号都移动到最左侧，把字母移到最右侧并保持相对顺序不变），要求时间和空间复杂度最小

10、时间复杂度为  $O(1)$ ，怎么找出一个栈里的最大元素

11、线程、进程区别

12、static 在 C 和 C++里各代表什么含义

13、const 在 C/C++里什么意思

14、常用 linux 命令

15、解释 Select/Poll 模型

6. 网易有道二面：

判断一个数字序列是 BST 后序遍历的结果，现场写代码。

来源：<http://blog.csdn.net/hopeztm/article/category/1201028>;

7. 8 月 30 日，网易有道面试题

```
var tt = 'aa';
function test()
{
    alert(tt);
    var tt = 'dd';
    alert(tt);
}
test();
```

8. 8月31日，百度面试题：不使用随机数的洗牌算法，详情：

<http://topic.csdn.net/u/20120831/10/C837A419-DFD4-4326-897C-669909BD2086.html> ;

9. 9月6日，阿里笔试题：平面上有很多点，点与点之间有可能有连线，求这个图里环的数目。

10. 9月7日，一道华为上机题：

题目描述：选秀节目打分，分为专家评委和大众评委，`score[]` 数组里面存储每个评委打的分数，`judge_type[]` 里存储与 `score[]` 数组对应的评委类别，`judge_type == 1`，表示专家评委，`judge_type == 2`，表示大众评委，`n` 表示评委总数。打分规则如下：专家评委和大众评委的分数先分别取一个平均分（平均分取整），然后，总分 = 专家评委平均分 \* 0.6 + 大众评委 \* 0.4，总分取整。如果没有大众评委，则 总分 = 专家评委平均分，总分取整。函数最终返回选手得分。

函数接口 `int cal_score(int score[], int judge_type[], int n)`

上机题目需要将函数验证，但是题目中默认专家评委的个数不能为零，但是如何将这种专家数目为 0 的情形排除出去。

来源：

<http://topic.csdn.net/u/20120907/15/c30eead8-9e49-41c2-bd11-c277030ad17a.html>;

11. 9月8日，腾讯面试题：

假设两个字符串中所含有的字符和个数都相同我们就叫这两个字符串匹配，

比如：`abcda` 和 `adabc`，由于出现的字符个数都是相同，只是顺序不同，

所以这两个字符串是匹配的。要求高效！

又是跟上述第 3 题中简单题一的兄弟节点类似的一道题，我想，你们能想到的，这篇 [blog](http://blog.csdn.net/v_JULY_v/article/details/6347454) 里：[http://blog.csdn.net/v\\_JULY\\_v/article/details/6347454](http://blog.csdn.net/v_JULY_v/article/details/6347454) 都已经有了。

12. 阿里云，搜索引擎中 5 亿个 url 怎么高效存储；

13. 一道 C++ 笔试题，求矩形交集的面积：

在一个平面坐标系上，有两个矩形，它们的边分别平行于 X 和 Y 轴。

其中，矩形 A 已知，`ax1` (左边)，`ax2` (右边)，`ay1` (top 的纵坐标)，`ay2` (bottom 纵坐标)。矩形 B，类似，就是 `bx1`，`bx2`，`by1`，`by2`。这些值都是整数就 OK 了。

要求是，如果矩形没有交集，返回 -1，有交集，返回交集的面积。

```
int area(rect const& a, rect const& b)
```

```
{  
    ...  
}
```

点评：

healer\_kx:

补齐代码，最好是简洁的，别用库。你可以写你的辅助函数，宏定义，代码风格也

很重要。

ri\_aje:

```
1. struct rect
2. {
3.     // axis alignment assumed
4.     // bottom left is (x[0],y[0]), top right is (x[1],y[1])
5.     double x [2];
6.     double y [2];
7. };
8.
9. template <typename T> T const& min (T const& x, T const& y) { return x<y ? x
    : y; }
10. template <typename T> T const& max (T const& x, T const& y) { return x>y ? x
    : y; }
11.
12. // return type changed to handle non-integer rects
13. double area (rect const& a, rect const& b)
14. {
15.     // perfectly adjacent rects are considered having an intersection of 0 area
16.     double const dx = min(a.x[1],b.x[1]) - max(a.x[0],b.x[0]);
17.     double const dy = min(a.y[1],b.y[1]) - max(a.y[0],b.y[0]);
18.     return dx>=0&&dy>=0 ? dx*dy : -1;
19. }
```

下面是一个简短的证明。

对于平行于坐标轴的矩形  $r$ ，假设其左下角点坐标为  $(rx0,ry0)$ ，右上角点坐标为  $(rx1,ry1)$ ，那么由  $r$  定义的无限有界点集为： $\{(x,y)|x \text{ in } [rx0,rx1] \ \&\& \ y \text{ in } [ry0,ry1]\}$ 。

根据交集的定义，则任意二维点  $(x,y)$  在矩形  $a,b$  的交集内等价于

$$\{(x,y)|(x,y) \text{ in } a \text{ 并且 } (x,y) \text{ in } b\} \iff$$
$$\{(x,y)|x \text{ in } [ax0,ax1] \ \&\& \ x \text{ in } [bx0,bx1] \text{ 并且 } y \text{ in } [ay0,ay1] \ \&\& \ y \text{ in } [by0,by1]\} \iff$$
$$\{(x,y)|x \text{ in } [\max(ax0,bx0),\min(ax1,bx1)] \text{ 并且 } y \text{ in } [\max(ay0,by0),\min(ay1,by1)]\}$$

因此，交集矩形的边长分别为  $\min(ax1,bx1)-\max(ax0,bx0)$  和  $\min(ay1,by1)-\max(ay0,by0)$ 。注意当交集为空时( $a,b$  不相交)，则经此法计算出来的交集边长为负值，此事实可用于验证  $a,b$  的相交性。

鉴于笛卡尔积各个维度上的不相关性，此方法可扩展到任意有限维线性空间，比如，三维空间中平行于坐标轴的长方体的交集体积可以用类似的方法计算。

来源：

<http://topic.csdn.net/u/20120913/18/bc669d60-b70a-4008-be65-7c342789b925.html>。

#### 14. 2012 年创新工场校园招聘最后一道笔试题：工场很忙

创新工场每年会组织同学与项目的双选会，假设现在有  $M$  个项目，编号从 1 到  $M$ ，另有  $N$  名同学，编号从 1 到  $N$ ，每名同学能选择最多三个、最少一个感兴趣的项目。选定之后，HR 会安排项目负责人和相应感兴趣的同学一对一面谈，每次面谈持续半小时。由于大家平时都很忙，所以咱们要尽量节约时间，请你按照以下的条件设计算法，帮助 HR 安排面试。

- 1) 同学很忙。项目负责人一次只能与一名同学面谈，而同学会在自己第一个面试开始时达到工场，最后一个面试结束后离开工场，如果参加一个项目组的面试后不能立即参加下一个项目组的面试，就必须在工场等待。所以请尽可能让同学的面试集中在某一时间段，减少同学在工场等待的时间。
- 2) 项目负责人很忙。众所周知，创业团队的负责人会有很多事情要做，所以他们希望能够将自己参与的面试集中在某一段时间内，请在保证 1) 的情况下，使得项目负责人等待的时间最少。
- 3) HR 很忙。从第一轮面试开始以后，所有 HR 都必须等到最后一轮面试结束，所以需要在保证 1) 和 2) 的同时，也能尽快解放掉所有的 HR，即让第一轮面试到最后一轮面试之间持续的时间最短。

**输入(以文件方式输入，文件名为 iw，例如 iw.in):**

第 1 行...第  $n$  行：同学的编号 项目的编号

样例（数据间用空格隔开,两个 0 表示输入结束）：

```
1 1
1 2
1 3
2 1
3 1
3 2
0 0
```

表示  $M=3$ ， $N=3$ ，编号为 1 的同学选择了项目 1,2 和 3，编号为 2 的同学选择了项目 1，编号为 3 的同学选了项目 1 和 2

**输出(以文件方式输出，文件名为 iw，例如 iw.out):**

第 1 行：编号为 1 的项目依次面试新同学的编号序列

第 2 行：编号为 2 的项目依次面试新同学的编号序列

...

第  $n$  行：编号为  $n$  的项目依次面试新同学的编号序列

样例（数据间用空格隔开，0 表示没有面试）：

```
1 3 2
3 1 0
0 0 1
```

表示编号为 1 的项目在第一轮面试编号为 1 的同学，第二轮面试编号为 3 的同学，第三轮面试编号为 2 的同学

编号为 2 的项目在第一轮面试编号为 3 的同学，第二轮面试编号为 1 的同学，第二轮不用面试

编号为 3 的项目在第一轮和第二轮都不用面试，第三轮面试编号为 1 的同学

链接: <http://t.qq.com/p/t/108332110988802>;

#### 15. 4\*\*9 的笔试题，比较简单:

- 1.求链表的倒数第二个节点
- 2.有一个整数数组，求数组中第二大的数

#### 16. 阿里巴巴二道题

第一道:

对于给定的整数集合  $S$ ，求出最大的  $d$ ，使得  $a+b+c=d$ 。 $a,b,c,d$  互不相同，且都属于  $S$ 。集合的元素个数小于等于 2000 个，元素的取值范围在  $[-2^{28}, 2^{28} - 1]$ ，假定可用内存空间为 100MB，硬盘使用空间无限大，试分析时间和空间复杂度，找出最快的解决方法。

点评:

@绿色夹克衫: 两两相加转为多项式乘法, 比如  $(1\ 2\ 4\ 6) + (2\ 3\ 4\ 5) \Rightarrow (x + x^2 + x^4 + x^6) * (x^2 + x^3 + x^4 + x^5)$ 。更多思路请见这:

<http://www.51nod.com/answer/index.html#!answerId=569>。

阿里巴巴第二道(研发类)

笔试题 1，原题大致描述有一大批数据，百万级别的。数据项内容是：用户 ID、科目 ABC 各自的成绩。其中用户 ID 为  $0 \sim 1000$  万之间，且是连续的，可以唯一标识一条记录。科目 ABC 成绩均在  $0 \sim 100$  之间。有两块磁盘，空间大小均为 512M，内存空间 64M。

- 1) 为实现快速查询某用户 ID 对应的各科成绩，问磁盘文件及内存该如何组织;
- 2) 改变题目条件，ID 为  $0 \sim 10$  亿之间，且不连续。问磁盘文件及内存该如何组织;
- 3) 在问题 2 的基础上，增加一个需求。在查询各科成绩的同时，获取该用户的排名，问磁盘文件及内存该如何组织。

笔试题 2：代码实现计算字符串的相似度。

点评：和计算两字符串的最长公共子序列相似。

设  $A_i$  为字符串  $A(a_1a_2a_3 \dots a_m)$  的前  $i$  个字符（即为  $a_1, a_2, a_3 \dots a_i$ ）

设  $B_j$  为字符串  $B(b_1b_2b_3 \dots b_n)$  的前  $j$  个字符（即为  $b_1, b_2, b_3 \dots b_j$ ）

设  $L(i, j)$  为使两个字符串和  $A_i$  和  $B_j$  相等的最小操作次数。

当  $a_i$  等于  $b_j$  时 显然  $L(i, j) = L(i-1, j-1)$

当  $a_i$  不等于  $b_j$  时



若将它们修改为相等，则对两个字符串至少还要操作  $L(i-1, j-1)$  次

若删除  $a_i$  或在  $B_j$  后添加  $a_i$ ，则对两个字符串至少还要操作  $L(i-1, j)$  次

若删除  $b_j$  或在  $A_i$  后添加  $b_j$ ，则对两个字符串至少还要操作  $L(i, j-1)$  次

此时  $L(i, j) = \min(L(i-1, j-1), L(i-1, j), L(i, j-1)) + 1$

显然， $L(i, 0) = i$ ， $L(0, j) = j$ ，再利用上述的递推公式，可以直接计算出  $L(i, j)$  值。具体代码请见这：<http://blog.csdn.net/flyinghearts/article/details/5605996>。

17. 9月14日，小米笔试，给一个浮点数序列，取**最大乘积子序列**的值，例如  $-2.5, 4, 0, 3, 0.5, 8, -1$ ，则取出的最大乘积子序列为  $3, 0.5, 8$ 。

点评：

解法一、

或许，读者初看此题，自然会想到最大乘积子序列问题类似于最大子数组和问题：[http://blog.csdn.net/v\\_JULY\\_v/article/details/6444021](http://blog.csdn.net/v_JULY_v/article/details/6444021)，然实则具体处理起来诸多不同，为什么呢，因为乘积子序列中有正有负也还可能有  $0$ 。

既如此，我们可以把问题简化成这样：数组中找一个子序列，使得它的乘积最大；同时找一个子序列，使得它的乘积最小（负数的情况）。因为虽然我们只要一个最大积，但由于负数的存在，我们同时找这两个乘积做起来反而方便。也就是说，不但记录最大乘积，也要记录最小乘积。So，

我们让 `maxCurrent` 表示当前最大乘积的 candidate，

`minCurrent` 反之，表示当前最小乘积的 candidate。

（用 `candidate` 这个词是因为只是可能成为新一轮的最大/最小乘积），

而 `maxProduct` 则记录到目前为止所有最大乘积 candidates 的最大值。

由于空集的乘积定义为  $1$ ，在搜索数组前，`maxCurrent`, `minCurrent`, `maxProduct` 都赋为  $1$ 。

假设在任何时刻你已经有了 `maxCurrent` 和 `minCurrent` 这两个最大/最小乘积的 candidates，新读入数组的元素  $x(i)$  后，新的最大乘积 candidate 只可能是 `maxCurrent` 或者 `minCurrent` 与  $x(i)$  的乘积中的较大者，如果  $x(i) < 0$  导致 `maxCurrent` < `minCurrent`，需要交换这两个 candidates 的值。

当任何时候 `maxCurrent` <  $1$ ，由于  $1$ （空集）是比 `maxCurrent` 更好的 candidate，所以更新 `maxCurrent` 为  $1$ ，类似的可以更新 `minCurrent`。任何时候 `maxCurrent` 如果比最好的 `maxProduct` 大，更新 `maxProduct`。

具体代码如下：

```
1. template <typename Comparable>
2. Comparable maxprod( const vector<Comparable>&v)
3. {
4.     int i;
```

```

5.     Comparable maxProduct = 1;
6.     Comparable minProduct = 1;
7.     Comparable maxCurrent = 1;
8.     Comparable minCurrent = 1;
9.     //Comparable t;
10.
11.     for( i=0; i< v.size() ;i++)
12.     {
13.         maxCurrent *= v[i];
14.         minCurrent *= v[i];
15.         if(maxCurrent > maxProduct)
16.             maxProduct = maxCurrent;
17.         if(minCurrent > maxProduct)
18.             maxProduct = minCurrent;
19.         if(maxCurrent < minProduct)
20.             minProduct = maxCurrent;
21.         if(minCurrent < minProduct)
22.             minProduct = minCurrent;
23.         if(minCurrent > maxCurrent)
24.             swap(maxCurrent,minCurrent);
25.         if(maxCurrent<1)
26.             maxCurrent = 1;
27.         //if(minCurrent>1)
28.         //    minCurrent =1;
29.     }
30.     return maxProduct;
31. }

```

解法二、

本题除了上述类似最大子数组和的解法，也可以直接用动态规划求解（其实，上述的解法一本质上也是动态规划，只是解题所表现出来的具体形式与接下来的解法二不同罢了。这个不同就在于下面的解法二会写出动态规划问题中经典常见的状态转移方程，而解法一是直接求解）。具体解法如下：

假设数组为  $a[]$ ，直接利用动归来求解，考虑到可能存在负数的情况，我们用  $Max[i]$  来表示以  $a[i]$  结尾的最大连续子序列的乘积值，用  $Min[i]$  表示以  $a[i]$  结尾的最小的连续子序列的乘积值，那么状态转移方程为：

$$Max[i]=\max\{a[i], Max[i-1]*a[i], Min[i-1]*a[i]\};$$

$$Min[i]=\min\{a[i], Max[i-1]*a[i], Min[i-1]*a[i]\};$$

初始状态为  $Max[1]=Min[1]=a[1]$ 。代码如下：

```

1.  /*
2.   给定一个整数数组，有正有负数，0，正数组成,数组下标从 1 算起

```

```

3. 求最大连续子序列乘积，并输出这个序列，如果最大子序列乘积为负数，那么就输出-1
4. 用 Max[i]表示以 a[i]结尾乘积最大的连续子序列
5. 用 Min[i]表示以 a[i]结尾乘积最小的连续子序列 因为有复数，所以保存这个是必须的
6. */
7. void longest_multiple(int *a,int n){
8.     int *Min=new int[n+1]();
9.     int *Max=new int[n+1]();
10.    int *p=new int[n+1]();
11.    //初始化
12.    for(int i=0;i<=n;i++){
13.        p[i]=-1;
14.    }
15.    Min[1]=a[1];
16.    Max[1]=a[1];
17.    int max_val=Max[1];
18.    for(int i=2;i<=n;i++){
19.        Max[i]=max(Max[i-1]*a[i],Min[i-1]*a[i],a[i]);
20.        Min[i]=min(Max[i-1]*a[i],Min[i-1]*a[i],a[i]);
21.        if(max_val<Max[i])
22.            max_val=Max[i];
23.    }
24.    if(max_val<0)
25.        printf("%d",-1);
26.    else
27.        printf("%d",max_val);
28.    //内存释放
29.    delete [] Max;
30.    delete [] Min;
31. }

```

## 变种

此外，此题还有另外的一个变种形式，即给定一个长度为 N 的整数数组，只允许用乘法，不能用除法，计算任意 (N-1) 个数的组合中乘积最大的一组，并写出算法的时间复杂度。

我们可以把所有可能的 (N-1) 个数的组合找出来，分别计算它们的乘积，并比较大小。由于总共有 N 个 (N-1) 个数的组合，总的时间复杂度为  $O(N^2)$ ，显然这不是最好的解法。

OK，以下解答来自编程之美

解法 1

在计算机科学中，时间和空间往往是一对矛盾体，不过，这里有一个折中方法。可以通过“空间换时间”或“时间换空间”的策略来达到优化某一方面的目的。在这里，是否可以通过“空间换时间”来降低时间复杂度呢？

计算  $(N-1)$  个数的组合乘积，假设第  $i$  个  $(0 \leq i \leq N-1)$  元素被排除在乘积之外（如图 2-16 所示）。

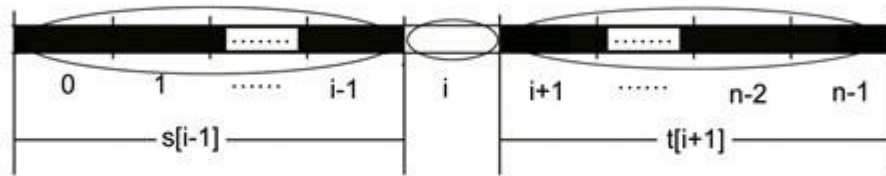


图 2-16 组合示意图。

设  $array[]$  为初始数组， $s[i]$  表示数组前  $i$  个元素的乘积  $s[i] = \prod_{j=1}^i array[j-1]$ ，其中  $1 \leq i \leq N$ ， $s[0] = 1$ （边界条件），那么  $s[i] = s[i-1] \times array[i-1]$ ，其中  $i = 1, 2, \dots, N-1, N$ ；

设  $t[i]$  表示数组后  $(N-i)$  个元素的乘积  $t[i] = \prod_{j=i}^N array[j]$ ，其中  $1 \leq i \leq N$ ， $t[N+1] = 1$ （边界条件），那么  $t[i] = t[i+1] \times array[i]$ ，其中  $i = 1, 2, \dots, N-1, N$ ；

那么设  $p[i]$  为数组除第  $i$  个元素外，其他  $N-1$  个元素的乘积，即有：

$$p[i] = s[i-1] \times t[i+1]。$$

由于只需要从头至尾，和从尾至头扫描数组两次即可得到数组  $s[]$  和  $t[]$ ，进而线性时间可以得到  $p[]$ 。所以，很容易就可以得到  $p[]$  的最大值（只需遍历  $p[]$  一次）。总的时间复杂度等于计算数组  $s[]$ 、 $t[]$ 、 $p[]$  的时间复杂度加上查找  $p[]$  最大值的时间复杂度等于  $O(N)$ 。

## 解法 2

此外，还可以通过分析，进一步减少解答问题的计算量。假设  $N$  个整数的乘积为  $P$ ，针对  $P$  的正负性进行如下分析（其中， $A_{N-1}$  表示  $N-1$  个数的组合， $P_{N-1}$  表示  $N-1$  个数的组合的乘积）。

### 1. $P$ 为 0

那么，数组中至少包含有一个 0。假设除去一个 0 之外，其他  $N-1$  个数的乘积为  $Q$ ，根据  $Q$  的正负性进行讨论：

#### $Q$ 为 0

说明数组中至少有两个 0，那么  $N-1$  个数的乘积只能为 0，返回 0；

#### $Q$ 为正数

返回  $Q$ ，因为如果以 0 替换此时  $A_{N-1}$  中的任一个数，所得到的  $P_{N-1}$  为 0，必然小于  $Q$ ；

#### $Q$ 为负数

如果以 0 替换此时  $A_{N-1}$  中的任一个数，所得到的  $P_{N-1}$  为 0，大于  $Q$ ，乘积最大值为 0。

## 2. $P$ 为负数

根据“负负得正”的乘法性质，自然想到从  $N$  个整数中去掉一个负数，使得  $P_{N-1}$  为一个正数。

而要使这个正数最大，这个被去掉的负数的绝对值必须是数组中最小的。我们只需要扫描一遍数组，把绝对值最小的负数给去掉就可以了。

## 3. $P$ 为正数

类似地，如果数组中存在正数值，那么应该去掉最小的正数值，否则去掉绝对值最大的负数值。

上面的解法采用了直接求  $N$  个整数的乘积  $P$ ，进而判断  $P$  的正负性的办法，但是直接求乘积在编译环境下往往会有溢出的危险（这也就是本题要求不使用除法的潜在用意），事实上可做一个小的转变，不需要直接求乘积，而是求出数组中正数（+）、负数（-）和 0 的个数，从而判断  $P$  的正负性，其余部分与上面的解法相同。

在时间复杂度方面，由于只需要遍历数组一次，在遍历数组的同时就可得到数组中正数（+）、负数（-）和 0 的个数，以及数组中绝对值最小的正数和负数，时间复杂度为  $O(N)$ 。

18. 9 月 15 日，中兴面试：

小端系统

```
1. union{
2.     int i;
3.     unsigned char ch[2];
4. }Student;
5.
6.
7. int main()
8. {
9.     Student student;
10.    student.i=0x1420;
11.    printf("%d %d",student.ch[0],student.ch[1]);
12.    return 0;
13. }
```

输出结果为？（答案：32 20）

### 19. 一道有趣的 Facebook 面试题：

给一个二叉树，每个节点都是正或负整数，如何找到一个子树，它所有节点的和最大？

点评：

☞某猛将兄：后序遍历，每一个节点保存左右子树的和加上自己的值。额外一个空间存放最大值。

☞陈利人：同学们，如果你面试的是软件工程师的职位，一般面试官会要求你在短时间内写出一个比较整洁的，最好是高效的，没有什么 bug 的程序。所以，光有算法不够，还得多实践。

写完后序遍历，面试官可能接着与你讨论，a). 如果要求找出只含正数的最大子树，程序该如何修改来实现？b). 假设我们将子树定义为它和它的部分后代，那该如何解决？c). 对于 b，加上正数的限制，方案又该如何？总之，一道看似简单的面试题，可能变换成各种花样。

比如，面试官可能还会再提两个要求：第一，不能用全局变量；第二，有个参数控制是否要只含正数的子树。其它的，随意，当然，编程风格也很重要。

### 20. 谷歌面试题：

有几百亿的整数，分布的存储到几百台通过网络连接的计算机上，你能否开发出一个算法和系统，找出这几百亿数据的中值？就是在排序好的数据中居于中间的数。显然，一台机器是装不下所有的数据。也尽量少用网络带宽。

### 21. 小米，南京站笔试（原第 20 题）：

一个数组里，数都是两两出现的，但是有三个数是唯一出现的，找出这三个数。

点评：

3 个数唯一出现，各不相同。由于  $x$  与  $a$ 、 $b$ 、 $c$  都各不相同，因此  $x^a$ 、 $x^b$ 、 $x^c$  都不等于 0。具体答案请参看这两篇文章：1、

<http://blog.csdn.net/w397090770/article/details/8032898>，2、

<http://zhedahht.blog.163.com/blog/static/25411174201283084246412/>。

### 22. 9 月 19 日，IGT 面试：你走到一个分叉路口，有两条路，每个路口有一个人，一个说假话，一个说真话，你只能问其中一个人仅一个问题，如何问才能得到正确答案？

点评：答案是，问其中一个人：另一个人会说你的路口是通往正确的道路么？

### 23. 9 月 19 日，创新工厂笔试题：

给定一整型数组，若数组中某个下标值大的元素值小于某个下标值比它小的元素值，称这是一个反序。

即：数组  $a[]$ ；对于  $i < j$  且  $a[i] > a[j]$ ，则称这是一个反序。

给定一个数组，要求写一个函数，计算出这个数组里所有反序的个数。

点评：

归并排序，至于有的人说是否有  $O(N)$  的时间复杂度，我认为答案是否定的，正如老梦所说，下限就是  $n \lg n$ ， $n$  个元素的数组的排列共有的排列是  $n \lg n$ ， $n!$ （算法导论里面也用递归树证明了： $O(n \cdot \log n)$ 是最优的解法，具体可以看下这个链接：）。

然后，我再给一个链接，这里有那天笔试的两道题目：

<http://blog.csdn.net/luno1/article/details/8001892>。

24. 9月20日，创新工厂南京站笔试：

已知字符串里的字符是互不相同的，现在任意组合，比如 **ab**，则输出 **aa**，**ab**，**ba**，**bb**，编程按照字典序输出所有的组合。

点评：非简单的全排列问题（跟全排列的形式不同，**abc** 全排列的话，只有 6 个不同的输出：

[http://blog.csdn.net/v\\_july\\_v/article/details/6879101](http://blog.csdn.net/v_july_v/article/details/6879101)）。本题可用递归的思想，设置一个变量表示已输出的个数，然后当个数达到字符串长度时，就输出。

```
1. //假设 str 已经有序，from 一直很安静
2. void perm(char *str, int size, int resPos)
3. {
4.     if(resPos == size)
5.         print(result);
6.     else
7.     {
8.         for(int i = 0; i < size; ++i)
9.         {
10.            result[resPos] = str[i];
11.            perm(str, size, resPos + 1);
12.        }
13.    }
14. }
```

25. 9月21日，小米，电子科大&西安交通大学笔试题：

1. 括号序列 由(){}[]组成 例如(([]{}))[] 这样的序列是合法的 {}{}或者{}{}或者{}{}就是不合法的序列。要求实现：  
(1).判断一个括号序列是否合法  
boolean isValidSeq(String input){}  
  
(2).如果一个序列不合法 请加入最少的括号数，将这个序列变成合法的。  
String fixSeq(String input){} （20分）

```
1. void fun()
2. {
3.     unsigned int a = 2013;
4.     int b = -2;
5.     int c = 0;
6.     while (a + b > 0)
7.     {
8.         a = a + b;
9.         c++;
10.    }
```



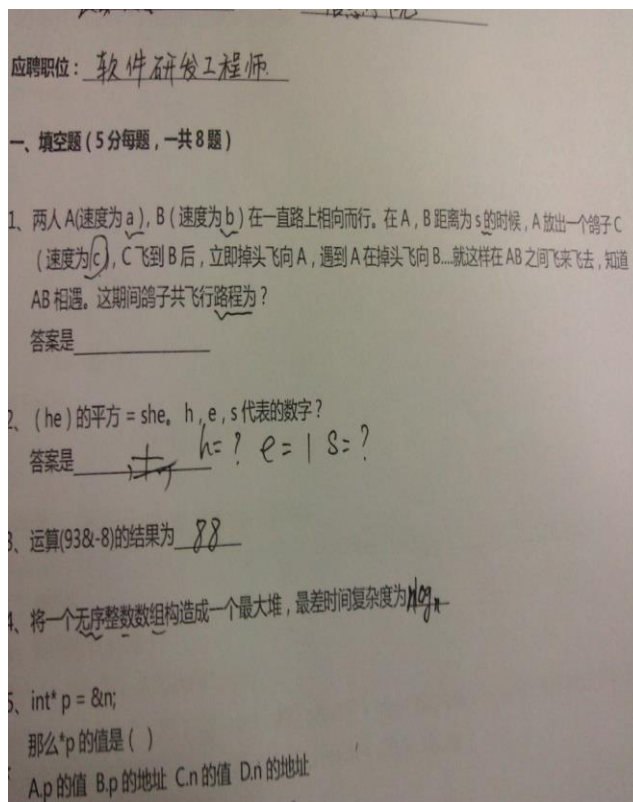
```

10.     }
11.     printf("%d", c);
12. }

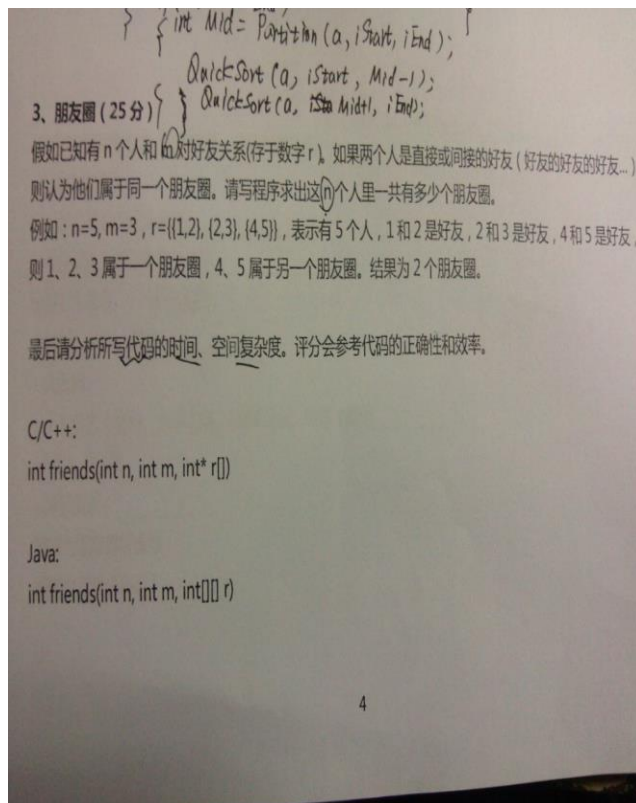
```

问：最后程序输出是多少？点评：此题有陷阱，答题需谨慎！

3, 给定一个浮点数的序列,  $F_1, F_2, F_3, \dots, F_n$  ( $1 \leq n \leq 1000$ ), 定义  $P(s, e)$  ( $1 \leq s \leq e \leq n$ ) 子序列  $F_i$  ( $s \leq i \leq e$ ) 的所有元素的乘积。写一个函数, 求  $P$  的最大值。输入保证对任意  $s, e$ ,  $P$  不会过  $\text{double}$  能表示的数据范围。(30分)  
比如序列:







点评:

针对上述第 3 题朋友圈的问题, 读者@互联网的飞虫提供的解法及代码如下(有任何问题, 欢迎指正, 多谢):

```

1. #include <STDIO.H>
2. #include <WINDOWS.H>
3.
4.
5. int Friends(int n, int m, int* r[]);
6.
7. int main(int argc, char** argv)
8. {
9.     int r[5][2] = {{1,2},{4,3},{6,5},{7,8},{7,9}};
10.
11.     printf("有%d 个朋友圈。 \n", Friends(0,5,(int**)r));
12.     return 0;
13. }
14.
15. int Friends(int n, int m, int* r[]) // 注意这里的参数很奇葩
16. {
17.
18.     int *p = (int*)malloc(sizeof(int)*m*3);
19.
20.     memset(p,0,sizeof(int)*m*3);

```

```
21.  int i = 0;
22.
23.  int iCount = 0;
24.
25.  int j = 0;
26.
27.  int * q = (int*)r; // 这里很巧妙 将二维指针 强转为一维指针
28.
29.  for (i=0;i<m;++i)
30.  {
31.      for (j=0;j<2;++j)
32.      {
33.          p[i*3+j]=q[i*2+j];    // 注意这里二维数组向一维数组的转换
34.      }
35.
36.      p[i*3+j] = 0;
37.  }
38.
39.  bool bFlag = false;
40.
41.  for (i=0;i<m;++i)
42.  {
43.      bFlag = false;
44.      if (p[i*3+2]==1)
45.      {
46.          bFlag = true;
47.      }
48.      p[i*3+2] = 1;
49.      for (j=0;j<m;++j)
50.      {
51.          if (i==j)
52.          {
53.              continue;
54.          }
55.
56.
57.          if (p[i*3]==p[j*3] ||
58.              p[i*3] == p[j*3+1] ||
59.              p[i*3+1] == p[j*3+0] ||
60.              p[i*3+1] == p[j*3+1])
61.          {
62.              if (p[j*3+2]==1)
63.              {
64.                  bFlag = true;
```

```

65.     }
66.     p[j*3+2] = 1;
67.     }
68. }
69.
70.     if (!bFlag)
71.     {
72.         ++iCount;
73.     }
74. }
75.
76. free(p);
77.
78. return iCount;
79. }

```

26. 9月21日晚，海豚浏览器笔试题：

1、有两个序列 A 和 B,  $A=(a_1,a_2,...,a_k)$ ,  $B=(b_1,b_2,...,b_k)$ , A 和 B 都按升序排列，对于  $1 \leq i,j \leq k$ ，求 k 个最小的  $(a_i+b_j)$ ，要求算法尽量高效。

2、输入：

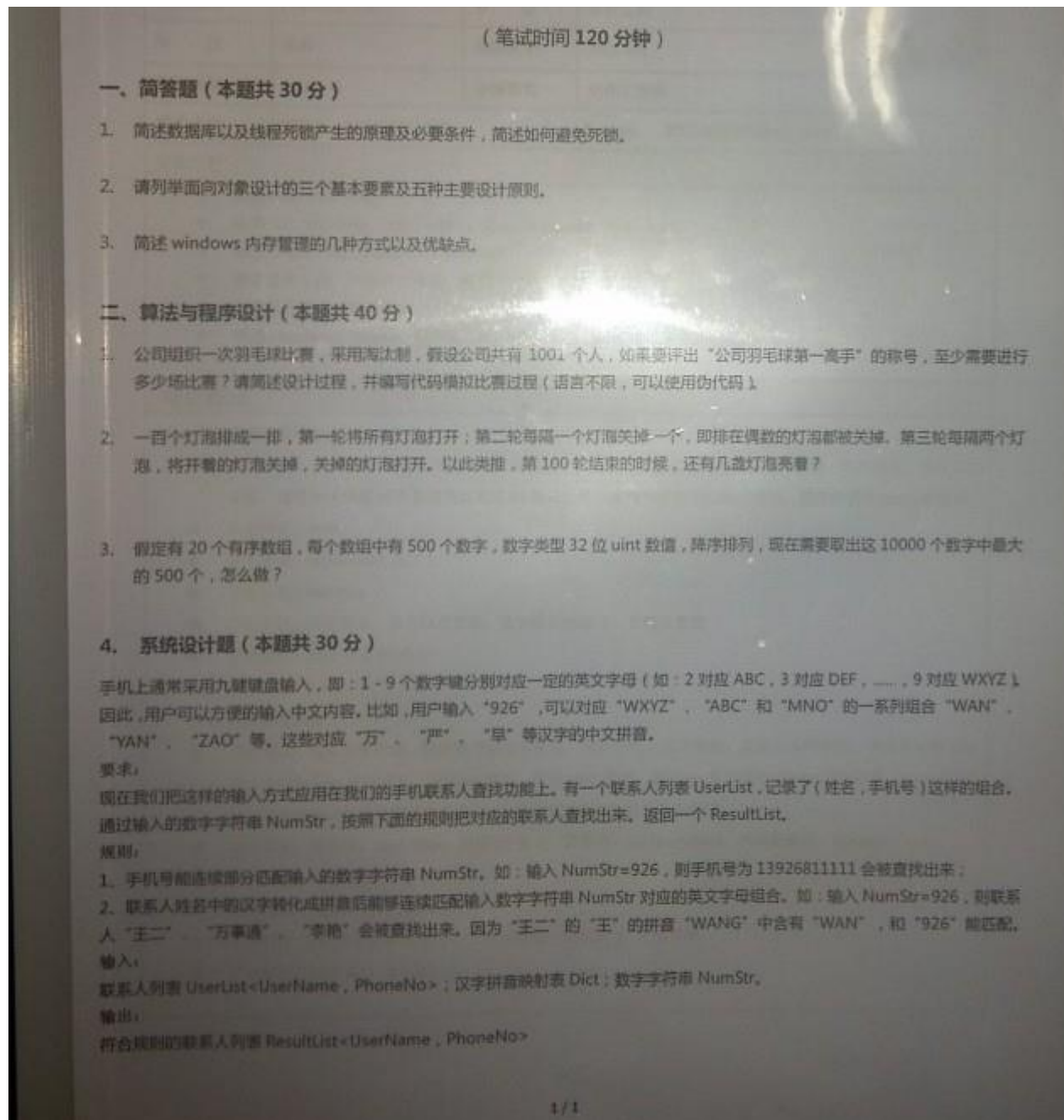
L: "shit" "fuck" "you"

S: "shitmeshitfuckyou"

输出：S 中包含的 L 一个单词，要求这个单词只出现一次，如果有多个出现一次的，输出第一个这样的单词

怎么做？

27. 9月22日上午，百度西安站全套笔试题如下：



点评：上述的系统设计题简单来讲，是建立起按键号码数字到人名(手机号)的映射关系，具体讲，步骤解法如下图所示：

- 1、手机号匹配部分：通过 KMP 等算法做快速的字符串匹配，将手机号符合条件的数据拉取出来，形成 ResultList1。
- 2、联系人部分：将联系人的姓名通过汉字拼音映射表，转换成拼音的列表。再将拼音按数字字母的映射关系数字字符串，做归一化。通过 KMP 等快速匹配算法，将姓名符合条件的数据拉取出来，形成 ResultList2。
- 3、将 ResultList1 和 ResultList2 做归并排序，形成最后的 ResultList。

### 3.算法与程序设计

第一题:

某个公司举行一场羽毛球赛,有 1001 个人参加,现在为了评比出“最厉害的那个人”,进行淘汰赛,请问至少需要进行多少次比赛。

第二题

有 100 个灯泡,第一轮把所有灯泡都开启,第二轮把奇数位的灯泡灭掉,第三轮每隔两个灯泡,灭一个,开一个,依此类推。求 100 轮后还亮的灯泡。

点评:完全平方数,本人去 58 面试时,也遇到过与此类似的题。

第三题

有 20 个数组,每个数组里面有 500 个数组,降序排列,每个数字是 32 位的 unit,求出这 10000 个数字中最大的 500 个。

点评: <http://www.51nod.com/question/index.html#!questionId=647>。

4.系统设计题

类似做一个手机键盘,上面有 1 到 9 个数字,每个数字都代表几个字母(比如 1 代表 abc 三个字母,z 代表 wxyz 等等),现在要求设计当输入某几个数字的组合时,查找出通讯录中的人名及电话号码。

其它的还有三道简答题,比如线程的死锁,内存的管理等等。最后,附一讨论帖子:

<http://topic.csdn.net/u/20120923/18/7fd148b2-c000-4326-93a6-cb3bb8675702.html>。

28. 9 月 22 日,微软笔试:

$T(n)=1(n \leq 1), T(n) = 25 * T(n/5) + n^2$ ,求算法的时间复杂度。更多题目请参见:

<http://blog.csdn.net/wonderwander6642/article/details/8008209>。

29. 9 月 23 日,腾讯校招部分笔试题(特别提醒:下述试卷上的答案只是一考生的解答,非代表正确答案.如下面第 11 题答案选 D,第 12 题答案选 C,至于解释可看这

里:<http://coolshell.cn/articles/7965.html>):

11. 题目：请问下面的程序一共输出多少个“-”？

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    int i;
    for(i=0; i<2; i++){
        fork();
        printf("-");
    }
    return 0;
}
```

A. 2个    B. 4个    C. 6个    D. 8个

12. 题目：请问下面的程序一共输出多少个“-”？

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
    int i;
    for(i=0; i<2; i++){
        fork();
        printf("-\n");
    }
    return 0;
}
```

A. 2个    B. 4个    C. 6个    D. 8个

13. 避免死锁的一个著名的算法是 ( )  
 A. 先来先出法    B. 银行家算法    C. 优先级算法    D. 资源按序分配法

14. 你怎么理解的分配延迟(dispatch latency)  
 A. 分配器停止一个进程到开启另一个进程的时间    B. 处理器将一个文件写入磁盘的时间  
 C. 所有处理器占用的时间    D. 以上都不对

15. 以下哪一个不是进程的基本状态 ( )  
 A. 阻塞态    B. 执行态    C. 就绪态    D. 完成态

16. 假定我们有 3 个程序，每个程序花费 80% 的时间进行 I/O, 20% 的时间使用 CPU。每个程序启动时间和其需要使用进行计算的分钟数如下，不考虑进程切换时间：

程序编号	启动时间	需要 CPU 时间 (分钟)
1	00:00	3.5
2	00:10	2
3	00:15	1.5

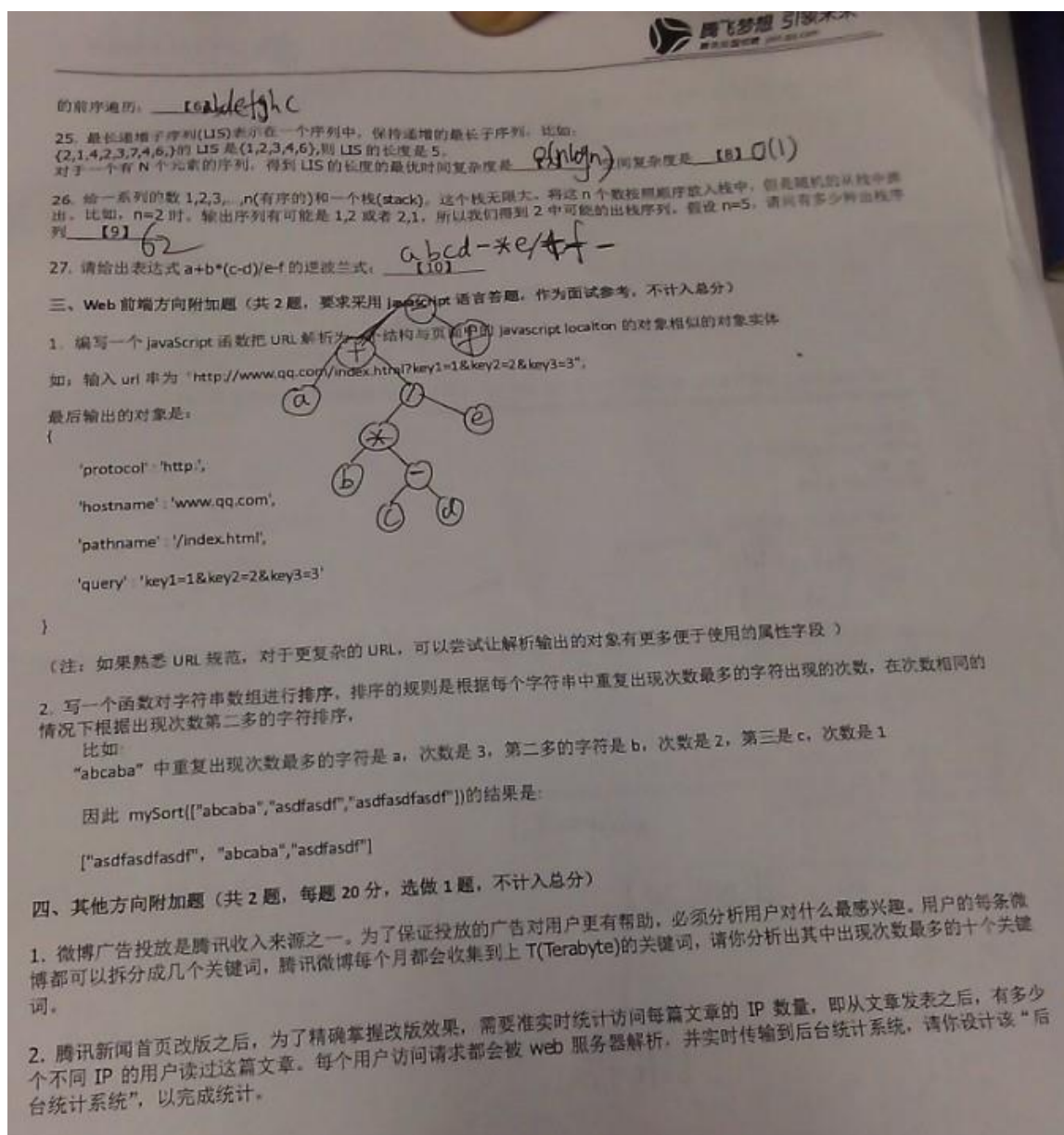
请问，在多线程/进程环境下，系统的总响应时间为 ( )  
 A. 22.5    B. 23.5    C. 24.5    D. 25.5

17. 在所有非抢占 CPU 调度算法中，系统平均响应时间最优的是 ( )  
 A. 实时调度算法    B. 短任务优先算法    C. 时间片轮转算法    D. 先来先服务算法

18. 什么是内存抖动(Thrashing) ( )  
 A. 非常频繁的换页活动    B. 非常高的 CPU 执行活动    C. 一个极长的执行进程    D. 一个极大的虚拟内存

19. Belady's Anomaly 出现在哪里 ( )  
 A. 内存管理算法    B. 内存换页算法    C. 预防锁死算法    D. 磁盘调度算法

20. 以下的生产者消费者程序中，那个不会出现锁死，并且开销最少？  
 注：  
 down()



点评: 根号九说, 不过最后两道大的附加题, 全是秒杀 99%海量数据处理面试题里的: [http://blog.csdn.net/v\\_july\\_v/article/details/7382693](http://blog.csdn.net/v_july_v/article/details/7382693), 太感谢 July 了。

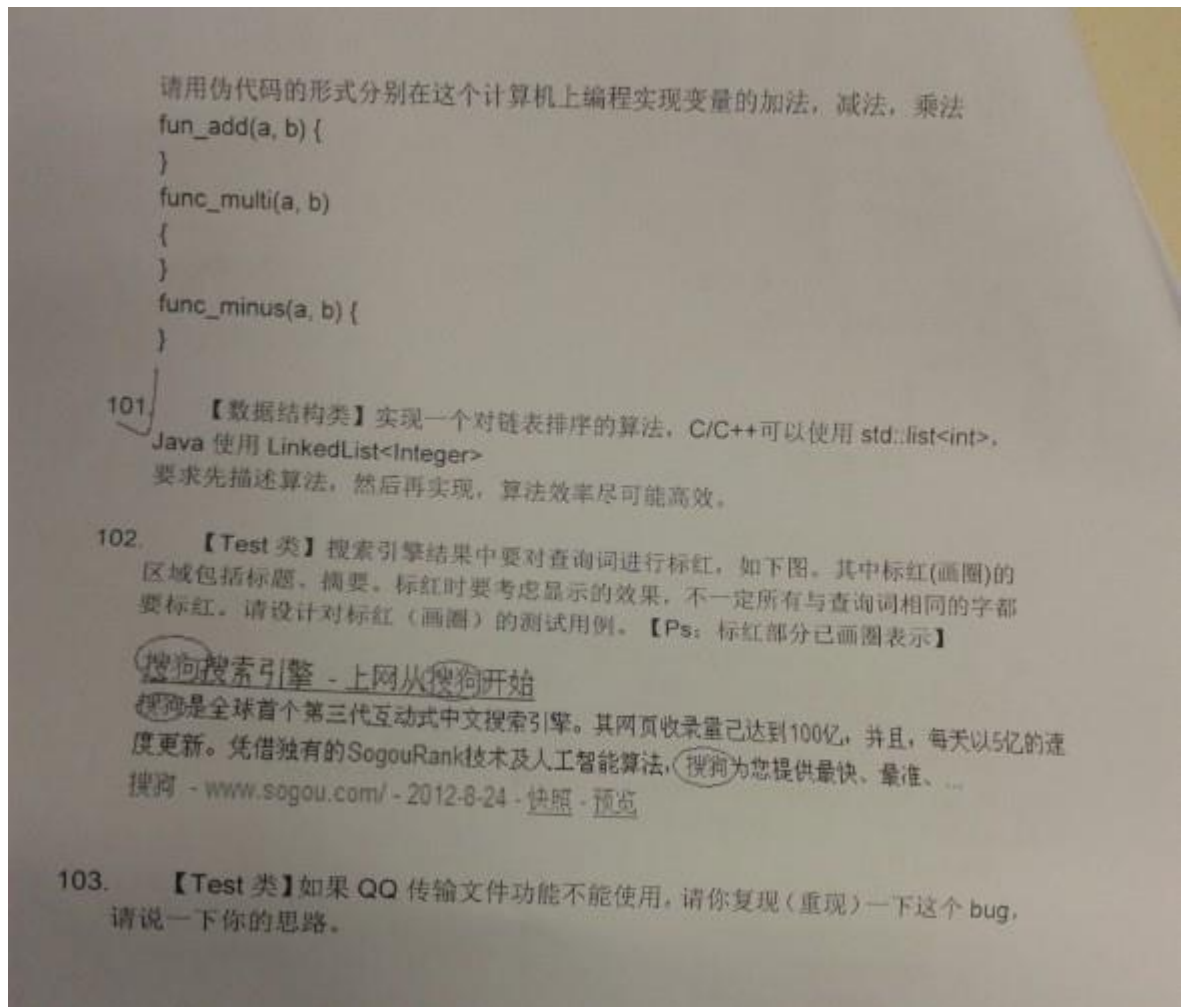
30. 9月23日, 搜狗校招武汉站笔试题:

一、已知计算机有以下原子操作

- 1、赋值操作:  $b = a$ ;
  - 2、 $++a$  和  $a+1$ ;
  - 3、 $\text{for}(\{ \} \{ \} \{ \})$  有限循环;
  - 4、操作数只能为 0 或者正整数;
  - 5、定义函数
- 实现加减乘操作



二、对一个链表进行排序，效率越高越好，LinkedList<Integer>.



附：9月15日，搜狐校招笔试题：

<http://blog.csdn.net/hackbuteer1/article/details/8015964>。

31. 搜狗校招笔试题：

100 个任务，100 个工人每人可做一项任务，每个任务每个人做的费用为

$t[100][100]$ ，求一个分配任务的方案使得总费用最少。

点评：匈牙利算法，可以看看这篇文章：<http://www.byvoid.com/blog/hungary/>，及这

个链接：<http://www.51nod.com/question/index.html#!questionId=641>。



32. 9月24日, Google 南京等站全套笔试题如下:

Google

2013 校园招聘笔试题

2013 校园招聘笔试题

姓名: \_\_\_\_\_ 手机号: \_\_\_\_\_ 分数: \_\_\_\_\_

1. 单项选择题

1.1. 使用 C 语言将一个 1G 字节的字符数组从头到尾全部设置为字符 'A', 在一台典型的当代 PC 上, 需要花费的 CPU 时间的数量级最接近:

A. 0.001 秒      B. 1 秒      C. 100 秒      D. 2 小时

1.2. 在某些极端要求性能的场景, 我们需要对程序进行优化, 关于优化, 以下说法正确的是:

A. 将程序整个用汇编语言改写会大大提高程序性能.  
B. 在优化前, 可以先确定哪部分代码最为耗时, 然后对这部分代码使用汇编语言改写, 使用的汇编语句数目越少, 程序就运行越快.  
C. 使用汇编语言虽然可能提高了程序性能, 但是降低了程序的可移植性和可维护性, 所以应该绝对避免.  
D. 适当调整汇编指令的顺序, 可以缩短程序运行的时间

1.3. 对如下 C 语言程序在普通 X86 PC 上面运行时候的输出叙述正确的是:

```
char *f()  
{  
    char X[512];  
  
    sprintf(X, "hello world");  
    return X+6;  
}  
main()  
{  
    printf("%s", f());  
}
```

A. 程序可能崩溃, 也可能输出 hello world  
B. 程序可能崩溃, 也可能输出 world  
C. 程序可能崩溃, 也可能输出 hello  
D. 程序一定会崩溃

1.4. 方程  $x_1+x_2+x_3+x_4=30$  有多少满足  $x_1>=2, x_2>=0, x_3>=-5, x_4>=8$  的整数解?

A. 3276      B. 3654      C. 2925      D. 17550

1.5. 一个袋子里装了 100 个苹果, 100 个香蕉, 100 个桔子, 100 个梨。如果每分钟

1

从里面随机抽取一个水果，那么最多过多少分钟时间能肯定至少拿到一打相同种类的水果？（1打=12个）

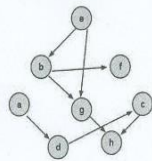
- A. 40      B. 12      C. 24      D. 45

1.6. 双败淘汰赛与淘汰赛相仿，也是负者出局，但负一场后并未被淘汰，只是跌入负者组，在负者组再负者（即总共已负两场）才被淘汰。现在有 10 个人参加双败淘汰赛，假设我们取消最后的胜者组冠军 VS 负者组冠军的比赛，那么一共需要举行多少场比赛？

- A. 16      B. 17      C. 18      D. 19      E. 20

1.7.  $n$  个节点的二叉树，最多可以有多少层？

- A.  $n/2$       B.  $\log(n)$       C.  $n-1$       D.  $n$



1.8.

下面哪个序列不是上图的一个拓扑排序？

- A. ebf gadch      B. aebdgfch      C. adchebfg      D. aedbfqch

1.9. 假如某主机安装了 2GB 内存，在其上运行的某支持 MMU 的 32 位 Linux 发行版中，一共运行了 X, Y, Z 三个进程。下面关于三个程序使用内存的方式，哪个是可行的？

- A. X, Y, Z 的虚拟地址空间都映射到 0~4G 的虚拟地址上  
B. X 在堆上分配总大小为 1GB 的空间，Y 在堆上分配 200MB，Z 在堆上分配 500MB，并且内存映射访问一个 1GB 的磁盘文件。  
C. X 在堆上分配 1GB，Y 在堆上分配 600MB，Z 在堆上分配 400MB  
D. 以上访问方式都是可行的

1.10. 当使用 TCP 协议编程时，下列问题哪个是必须由程序员考虑和处理的？

- A. 乱序数据包的重传      B. 数据传输过程中的纠错  
C. 网络拥塞处理      D. 发送数据的格式和应用层协议

## 2. 程序设计与算法

(2.1, 2.2 为编程题，需要写出程序实现；2.3 为算法设计题，只需要给出算法设计思路及关键步骤的伪代码即可。)

2.1. 给定三个整数  $a, b, c$ ，实现函数 `int median(int a, int b, int c)`，返回三个数的中位数。不可以使用 `sort`，要求整数操作（比较，位运算，加减乘除等）次数尽量少。并分析说明程序最坏和平均情况下使用的操作次数。

2.2. 给定一个 `key`（只含有 ASCII 编码的小写英文字母），例如 `kof`，然后对 `input` 的 `string`（只含有 ASCII 编码的小写英文字母）利用这个 `key` 排序。顺序是：先按照 `key` 中的字符顺序，然后对 `key` 中不包含的字符，按 `a-z` 顺序。

2.3. 一个平行于坐标轴的  $n \times n$  的网格，左下角  $(0, 0)$  右上角  $(n, n)$ ， $n$  为非负整数，有  $n$  个平行于坐标轴的矩形，每个矩形用左下角  $(x1, y1)$  右上角  $(x2, y2)$  来表示， $x1, y1, x2, y2$  都是非负整数。现在有非常多个 query，每个 query 会询问一个网格  $(x, y) (x+1, y+1)$  一共被几个矩形覆盖。现在要求设计一个算法，使得处理每个 query 的时间复杂度尽可能低。在这个前提下，预处理的时间复杂度尽可能低。

$1 \leq n \leq 1000$

点评：

谷歌的笔试从易到难，基础到复杂，涵盖操作系统 网络 数据结构 语言 数学思维 编程能力 算法能力，基本上能把一个人的能力全面考察出来。

至于上述 2.1 寻找 3 个数的中位数,请看读者 sos-phoenix 给出的思路及代码：

```
1. 2.1 // 采用两两比较的思路（目前没想到更好的）
2.     if (a <= b) {
3.         if (b <= c)
4.             return b;
5.         else {
6.             if (a <= c)
7.                 return c;
8.             else
9.                 return a;
10.        }
11.    }
12.    else {
13.        if (a <= c)
14.            return a;
15.        else {
16.            if (b <= c)
17.                return c;
18.            else
19.                return b;
20.        }
21.    }
```

最坏情况下的比较次数：3 （次）

平均情况下的比较次数： $(2 \times 2 + 4 \times 3) / 6 = 8 / 3$  （次）

此外这题，微博上的左耳朵耗子后来也给出了一个链

接：<http://stackoverflow.com/questions/1582356/fastest-way-of-finding-the-middle-value-of-a-triple>，最后是微博上的梁斌 penny 的解答：

<http://weibo.com/1497035431/yFusm7obQ>。其余更多参考答案请看本文评论下第 93 楼。

33. 读者来信，提供的几个 hulu 面试题：

9 月 19 号, hulu 电面：

问题 1 两个骰子，两个人轮流投，直到点数和大于 6 就停止，最终投的那个人获胜。

问先投那个人获胜概率？

问题 2 平面上  $n$  个圆，任意两个都相交，是否有一条直线和所有的圆都有交点。

9 月 22 号, 上午 hulu 面试

问题 1 100 个人，每人头上戴一顶帽子，写有  $0..99$  的一个数，数可能重复，每个人都只能看到除自己以外其他人的帽子。每个人需要说出自己的帽子的数，一个人说对就算赢。点评：参考答案请看这个链接：

<http://www.51nod.com/question/index.html#!questionId=642>。

问题 2  $n$  台机器，每台有负载，以和负载成正比的概率，随机选择一台机器。「原题是希望设计  $O(1)$  的算法（预处理  $O(n)$  不可少，要算出每台机器的比例），因为非  $O(1)$  的话，就 trivial 了：可以产生随机数例如  $[0,1)$  然后，根据负载比例，2 分或者直接循环检查落入哪个区间，决定机器。面试官想问，有没更好的办法，避免那种查找。即能否多次（常数级）调用随机函数，拟合出一个概率分布」

问题 3 行列都递增的矩阵，求中位数。点评：

<http://www.51nod.com/question/index.html#!questionId=643>,

[http://blog.csdn.net/v\\_july\\_v/article/details/7085669](http://blog.csdn.net/v_july_v/article/details/7085669)（杨氏矩阵查找问题）。

34. 西安百度软件研发工程师：

一面 (2012. 9. 24)：

问的比较广，涉及操作系统、网络、数据结构。比较难的就 2 道题。

(1) 10 亿个 int 型整数，如何找出重复出现的数字；

(2) 有 2G 的一个文本文件，文件每行存储的是一个句子，每个单词是用空格隔开的。

问：输入一个句子，如何找到和它最相似的前 10 个句子。（提示：可用倒排文档）。

二面 (2012. 9. 25)：

(1) 一个处理器最多能处理  $m$  个任务。现在有  $n$  个任务需要完成，每个任务都有自己完成所需的时间。此外每个任务之间有依赖性，比如任务 A 开始执行的前提是任务 B 必须完成。设计一个调度算法，使得这  $n$  个任务的完成时间最小；

(2) 有一个排序二叉树，数据类型是 int 型，如何找出中间大的元素；

(3) 一个  $N$  个元素的整形数组，如何找出前  $K$  个最大的元素。

(4) 给定一个凸四边形，如何判断一个点在这个平面上。

点评：本题的讨论及参考答案请见这：

<http://www.51nod.com/question/index.html#!questionId=669>。

运维部 (2012. 9. 27)：

- (1)堆和栈的区别;
- (2)问如何数出自己头上的头发。

35. 9月25日, 人人网笔试题:

4、七夕那天, 雯雯的男朋友小瑜给她买了一颗神奇的魔石。这颗魔石平常是黑的, 但只要给这颗魔石擦上魔粉, 魔石就会从内部发出不同颜色的绚丽的光泽, 非常漂亮。魔石发出的光的颜色, 是在魔石上擦的所有魔粉的编号的异或(xor)值。

(如果异或值为0, 也是一种颜色, 异或值相同表示颜色相同)。雯雯手上现在有6种魔粉, 编号是6、7、17、46、47、56。雯雯非常想知道, 这颗魔石究竟能发出多少种颜色的光。由于组合实在太多了, 雯雯便跑过来求助聪明的你, 相信你不会让她失望。

(1)请给出这6种魔粉一共能组成多少种颜色。

(2)如果这种魔粉有20种(编号在256以内), 有什么解题思路?

点评: 参考答案请见, <http://www.51nod.com/question/index.html#!questionId=671>。

36. 9月25日晚, 创新工场校园招聘北邮站笔试:

姓名: \_\_\_\_\_ 性别: \_\_\_\_\_ 学校: \_\_\_\_\_ 手机: \_\_\_\_\_ 毕业时间: \_\_\_\_\_ 年 月

工作地点: \_\_\_\_\_ 学历: \_\_\_\_\_ 邮箱: \_\_\_\_\_ 面试时间: \_\_\_\_\_

北京 ( ) 上海 ( ) 成都 ( ) 广州 ( )

第一部分 单选题, 把答案写在题目前括号中

( ) 1. 下述代码的输出结果(假定为32位机器)为( )。

```
struct s
{
    int x; 8;
    int y; 4;
    double a;
    int z; 3;
};
printf("%d\n", sizeof(s));
```

A. 8 B. 16 C. 20 D. 24

( ) 2. 一棵二叉树高度为h, 所有结点的度或为0, 或为2, 则这棵二叉树最少有( )结点

A. 2h B. 2h-1 C. 2h+1 D. h+1

( ) 3. 与10.110.12.29 mask 255.255.255.224属于同一网段的主机IP地址是( )。

A. 10.110.12.0 B. 10.110.12.30 C. 10.110.12.31 D. 10.110.12.32

( ) 4. 若系统中有五台绘图仪, 有多个进程均需要使用两台, 规定每个进程一次仅允许申请一台, 则至多允许( )个进程并发运行, 而不会发生死锁。

A. 2 B. 3 C. 4 D. 5

( ) 5. #define N 2  
#define Y(n) ((n+1)%n)  
则执行语句x=2 \* Y(5+1);后, x的值为( )。

A. 22 B. 35 C. 48 D. 出错

( ) 6. 2012! 的末尾有多少个0?

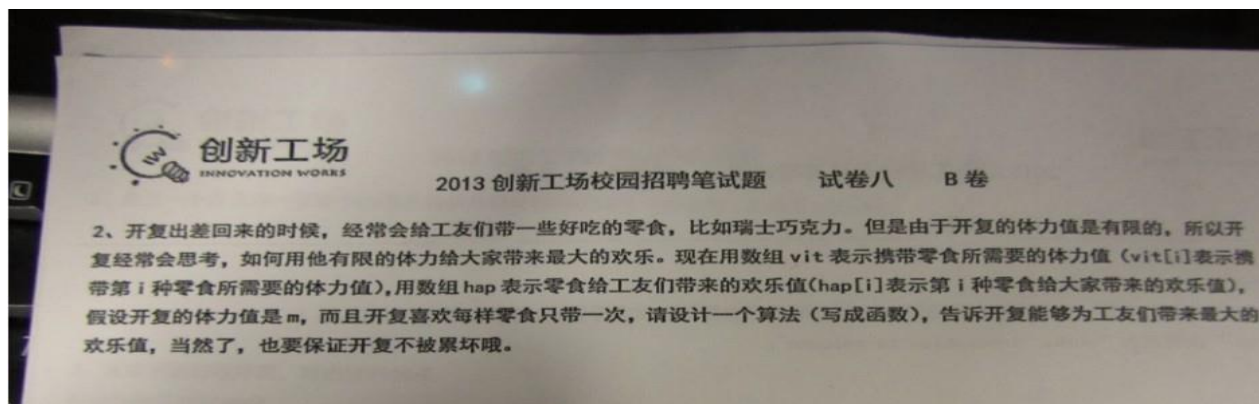
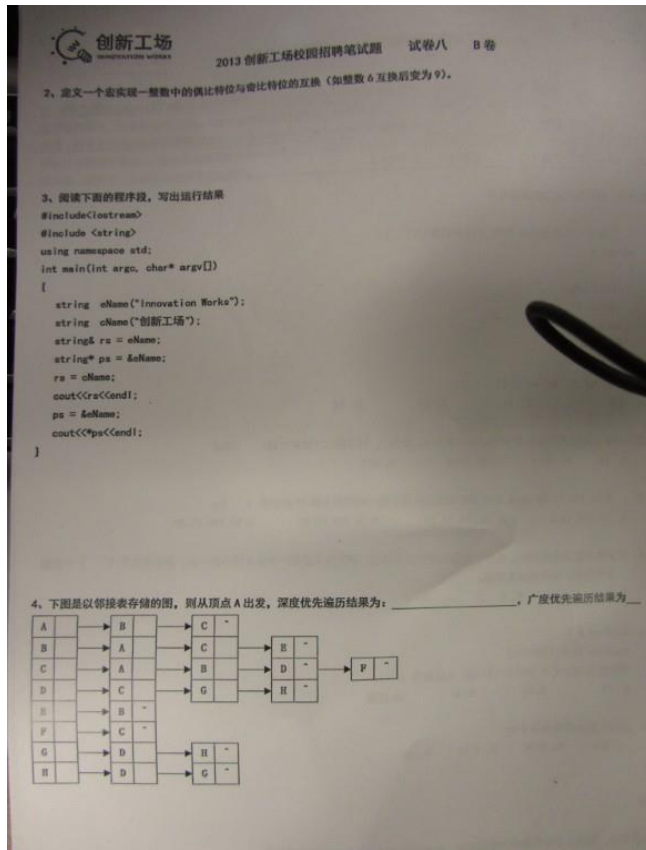
A. 501 B. 2004 C. 1994 D. 40

第二部分 简答题

1. 在百度或淘宝搜索时, 每输入字符都会出现搜索建议, 实现这类技术后台采用的数据结构是\_\_\_\_\_。

1/4





37. 9 月 25 日，小米大连站笔试题：

1 一共有 100 万，抽中的 2 万，每月增加 4 万，问 20 个月能抽中的概率为：？

2 for(int i=0;i<strlen(s);i++){n+=I;}时间复杂度  $O(n)$

3 手机 wifi (A) ... wifi ap ... 局域网 (B) ... 路由器 ... ADSL (C) ... 互联网 ... 服务器

断掉上述 ABC 哪些点 TCP 链接会立刻断掉?

4 12345 入栈, 出栈结果 21543 31245 43215 12534 可能的为? (第一个和第三个)

5  $x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$ , 最少要做一乘法? 题目中  $a_1, a_2, a_n$  为常数。

38. 9月26日, 百度一二面:

1、给定一数组, 输出满足  $2a=b$  ( $a, b$  代表数组中的数) 的数对, 要求时间复杂度尽量低。

2、搜索引擎多线程中每个线程占用多少内存? 如果搜索引擎存储网页内存占用太大怎么解决?

3、有很多 url, 例如 \*.baidu.com, \*.sina.com .....

现在给你一个 sports.sina.com 快速匹配出是 \*.sina.com。点评: 老题, 此前 blog 内曾整理过。

4、找出字符串的编辑距离, 即把一个字符串  $s_1$  最少经过多少步操作变成编程字符串  $s_2$ , 操作有三种, 添加一个字符, 删除一个字符, 修改一个字符 (只要听过编辑距离, 知道往动态规划上想, 很快就可以找到解法)。



新闻 网页 贴吧 知道 MP3 图片 视频 地图 hao123 更多»

结构执法

百度一下

推荐:

您要找的是不是: [结构之法](#)

[结构之法 算法之道 - 博客频道 - CSDN.NET](#)

[结构之法 算法之道](#)blog博文集锦第6、7期CHM文件0积分下载第6期CHM文件 新春回馈读者, 免积分下载本blog最新博文集锦第6期CHM文件(包含前五期的全部内容): [http...](#)

[blog.csdn.net/v\\_JULY\\_v 2012-10-9 - 百度快照](#)

点评: 请看链接: [http://blog.csdn.net/Lost\\_Painting/article/details/6457334](http://blog.csdn.net/Lost_Painting/article/details/6457334)。

5、编程实现 memcpy, 注意考虑目标内存空间和源空间重叠的时候。

6、实现简单的一个查找二叉树的深度的函数。

39. 9月26日晚, 优酷土豆笔试题一道:

优酷是一家视频网站, 每天有上亿的视频被观看, 现在公司要请研发人员找出最热门的视频。

该问题的输入可以简化为一个字符串文件, 每一行都表示一个视频 id, 然后要找出出现次数最多的前 100 个视频 id, 将其输出, 同时输出该视频的出现次数。

1. 假设每天的视频播放次数为 3 亿次, 被观看的视频数量为一百万个, 每个视频 ID

的长度为 20 字节，限定使用的内存为 1G。请简述做法，再写代码。

2. 假设每个月的视频播放次数为 100 亿次，被观看的视频数量为 1 亿，每个视频 ID 的长度为 20 字节，一台机器被限定使用的内存为 1G。

点评：有关海量数据处理的题目，请到此文中找方法（无论题目形式怎么变，基本方法不变，当然，最最常用的方法是：分而治之/Hash 映射 + Hash 统计 + 堆/快速/归并排序）：

[http://blog.csdn.net/v\\_july\\_v/article/details/7382693](http://blog.csdn.net/v_july_v/article/details/7382693)。注：上题第二问文件太大，则  
可如模 1000，把整个大文件映射为 1000 个小文件再处理 ....

#### 40. 9 月 26 日，baidu 面试题：

1. 进程和线程的区别
2. 一个有序数组（从小到大排列），数组中的数据有正有负，求这个数组中的最小绝对值
3. 链表倒数第 n 个元素
4. 有一个函数 fun 能返回 0 和 1 两个值，返回 0 和 1 的概率都是 1/2，问怎么利用这个函数得到另一个函数 fun2，使 fun2 也只能返回 0 和 1，且返回 0 的概率为 1/4，返回 1 的概率为 3/4。（如果返回 0 的概率为 0.3 而返回 1 的概率为 0.7 呢）
5. 有 8 个球，其中有 7 个球的质量相同，另一个与其他球的质量不同（且不知道是比其他球重还是轻），请问在最坏的情况下，最少需要多少次就能找出这个不同质量的球
6. 数据库索引
7. 有一个数组 a，设有一个值 n。在数组中找到两个元素 a[i] 和 a[j]，使得 a[i]+a[j] 等于 n，求出所有满足以上条件的 i 和 j。
8. 1 万个元素的数组，90% 的元素都是 1 到 100 的数，10% 的元素是 101—10000 的数，如何高效排序。

#### 41. 小米的 web 开发笔试题：

一场星际争霸比赛，共 8 个人，每个人的实力用分数表示，要分成两队，如何保证实力最平均？给定一个浮点数的序列，F1, F2, ……，Fn ( $1 \leq n \leq 1000$ )，定义 P(s, e) 为子序列 Fi ( $s \leq i \leq e$ ) 的积，求 P 的最大值。

#### 42. 9 月 27 日，趋势科技面试题：

马路口，30 分钟内看到汽车的概率是 95%，那么在 10 分钟内看不到汽车的概率是？

#### 43. 9 月 27 日晚，IGT 笔试题：

给定一个字符串里面只有“R” “G” “B” 三个字符，请排序，最终结果的顺序是 R 在前 G 中 B 在后。

要求：空间复杂度是 O(1)，且只能遍历一次字符串。

点评：本质是荷兰国旗问题，类似快排中 partition 过程，具体思路分析及代码可以参考此文第 8 节：[http://blog.csdn.net/v\\_july\\_v/article/details/6211155](http://blog.csdn.net/v_july_v/article/details/6211155)。



44. 9月27日，人人两面：

一面

- 1 实现 atoi
- 2 单链表变形 如 1 2 3 4 5 变为 1 3 5 4 2 如 1 2 3 4 变为 1 3 4 2  
(就是拆分链表 把偶数为反过来接在奇数位后面)

二面

- 1 二叉树查找不严格小于一个值的最大值（返回节点）。
- 2 有序数组里二分查找一个数（如果有相同的找最后一次出现的）。
- 3 等价于  $n \times n$  的矩阵，填写 0, 1，要求每行每列的都有偶数个 1（没有 1 也是偶数个），问有多少种方法。

评论：开始以为是算法题，想了狂搜，递推(dp，可以用 xor 表示一行的列状态，累加)，分治，（拆两半，然后上半段下半段的列有相同的奇偶性）。后来，自己算了几个发现  $n = 1$   $n = 2$   $n = 3$  的结果，他告诉了我  $n = 4$  是多少，然后发现  $f(n) = 2^{((n-1)^2)}$ 。最后我给出了一个巧妙的证明。然后发现如果是  $m \times n$  的矩阵也是类似的答案，不局限于方阵。此外，题目具体描述可以看看这里：

<http://blog.himdd.com/?p=2480>。

9月27日，小米两面：

一面：

除了聊研究，就一道题

- 1 数组里找到和最接近于 0 的两个值。

二面：

- 1 行列有序的矩阵查找一个数
- 2 直方图最大矩形。点评：这里有此题的具体表述及一份答案：

<http://blog.csdn.net/xybsos/article/details/8049048>。

- 3 next\_permutation
- 4 字符串匹配 含有\*？（写代码）
- 5 实现 strcpy memmove（必须写代码）

```
1. //void * memmove ( void * destination, const void * source, size_t num );  
2. //是<string.h>的标准函数，其作用是把从 source 开始的 num 个字符拷贝到 destination。  
3. //最简单的方法是直接复制，但是由于它们可能存在内存的重叠区，因此可能覆盖了原有数据。  
4. //比如当 source+count>=dest&&source<dest 时，dest 可能覆盖了原有 source 的数据。  
5. //解决办法是从后往前拷贝。  
6. //对于其它情况，则从前往后拷贝。  
7. void* memmove(void* dest, void* source, size_t count)  
8. {  
9.  
10. void* ret = dest;
```

```

11.
12.     if (dest <= source || dest >= (source + count))
13.     {
14.         //正向拷贝
15.         //copy from lower addresses to higher addresses
16.         while (count --)
17.             *dest++ = *source++;
18.     }
19.     else
20.     {
21.         //反向拷贝
22.         //copy from higher addresses to lower addresses
23.         dest += count - 1;
24.         source += count - 1;
25.
26.         while (count--)
27.             *dest-- = *source--;
28.     }
29.     return ret;
30. }

```

更多，还可以参见此文第三节节末：

[http://blog.csdn.net/v\\_july\\_v/article/details/6417600](http://blog.csdn.net/v_july_v/article/details/6417600)，或此

文：[http://www.360doc.com/content/11/0317/09/6329704\\_101869559.shtml](http://www.360doc.com/content/11/0317/09/6329704_101869559.shtml)。

6 读数（千万亿，百万亿……）变为数字（说思路即可，字符串查找，填写各个权值的字段，然后判断是否合法，读前面那些×权值，累加）。

#### 45. 9月27日，Hulu 2013 北京地区校招笔试题

填空题：

- 1、中序遍历二叉树，结果为 ABCDEFGH，后序遍历结果为 ABEDCHGF，那么前序遍历结果为？
- 2、对字符串 HELLO\_HULU 中的字符进行二进制编码，使得字符串的编码长度尽可能短，最短长度为？
- 3、对长度 12 的有序数组进行二分查找，目标等概率出现在数组的每个位置上，则平均比较次数为？
- 4、一副扑克（去王），每个人随机的摸两张，则至少需要多少人摸牌，才能保证有两个人抽到同样的花色。
- 5、x 个小球中有唯一一个球较轻，用天平秤最少称量 y 次能找出这个较轻的球，写出 y 和 x 的函数表达式  $y=f(x)$
- 6、3 的方幂及不相等的 3 的方幂的和排列成递增序列 1, 3, 4, 9, 10, 12, 13……，写出

数列第 300 项

7、无向图  $G$  有 20 条边，有 4 个度为 4 的顶点，6 个度为 3 的顶点，其余顶点度小于 3，则  $G$  有多少个顶点

8、桶中有  $M$  个白球，小明每分钟从桶中随机取出一个球，涂成红色（无论白或红都涂红）再放回，问小明将桶中球全部涂红的期望时间是？

9、煤矿有 3000 吨煤要拿到市场上卖，有一辆火车可以用来运煤，火车最多能装 1000 吨煤，且火车本身需要烧煤做动力，每走 1 公里消耗 1 吨煤，如何运煤才能使得运到市场的煤最多，最多是多少？

10、 $1, 2, 3, 4 \dots n$ ， $n$  个数进栈，有多少种出栈顺序，写出递推公式（写出通项公式不得分）

11、宇宙飞船有 100,000 位的存储空间，其中有一位有故障，现有一种 Agent 可以用来检测故障，每个 Agent 可以同时测试任意个位数，若都没有故障，则返回 OK，若有一位有故障，则失去响应。如果有无限多个 Agent 可供使用，每个 Agent 进行一次检测需要耗费 1 小时，现在有 2 个小时时间去找出故障位，问最少使用多少个 Agent 就能找出故障。

（总共 12 道填空题，还有一道太复杂，题目很长，还有示意图，这里没有记录下来）

大题：

1、 $n$  个数，找出其中最小的  $k$  个数，写出代码，要求最坏情况下的时间复杂度不能高于  $O(n \log k)$

2、写程序输出 8 皇后问题的所有排列，要求使用非递归的深度优先遍历

3、有  $n$  个作业， $a_1, a_2 \dots a_n$ ，作业  $a_j$  的处理时间为  $t_j$ ，产生的效益为  $p_j$ ，最后完成期限为  $d_j$ ，作业一旦被调度则不能中断，如果作业  $a_j$  在  $d_j$  前完成，则获得效益  $p_j$ ，否则无效益。给出最大化效益的作业调度算法。点评：参考答案请看这个链接：<http://www.51nod.com/question/index.html#!questionId=645>。

46. 有道的一个笔试题，1-9，9 个数组成三个三位数，且都是完全平方数（三个三位数占据 9 个数）求解法。

点评@林晚枫&归云见鸿：

$(a*10+b)(a*10+b)$

$100a^2+20ab+b^2$

$a$  属于  $[1,2,3]$

$a=3, b=1$  31 961,

$a=2, b=3$  23 529  $400+40b+b^2$

25 625

27 729

28 784

29 841  
a=1,b=3 13 169 100+20b+b^2  
14 196  
16 256  
17 289  
18 324  
19 361

=>最终唯一解 529 784 361

具体代码如下（3 个 for 循环，然后 hash）：

```
for(int i=13;i<=31;i++)
    for(int j=i+1;j<=31;j++)
        for(int k=j+1;k<=31;k++)
        {
            for(int u=0;u<=9;u++)
                hash[u]=0;
            int isfind=1;
            int a = i*i;
            int b = j*j;
            int c = k*k;
            while (a!=0)
            {
                if (a%10 == 0 || hash[a%10] != 0)
                {
                    isfind=0;
                    break;
                }
                else
                {
                    isfind=1;
                    hash[a%10]=1;
                    a=a/10;
                }
            }
        }
```

47. 9 月 28 日，大众点评北京笔试题目：

1. 一个是跳台阶问题，可以 1 次一级，1 次两级，1 次三级，求 N 级的跳法一共多少种？

点评：老题，参考答案请见：[http://blog.csdn.net/v\\_july\\_v/article/details/6879101](http://blog.csdn.net/v_july_v/article/details/6879101)。

2. 一个文件有 N 个单词，每行一个，其中一个单词出现的次数大于 N/2，怎么样才能快速找出这个单词？

点评：还是老题，参见：[http://blog.csdn.net/v\\_july\\_v/article/details/6890054](http://blog.csdn.net/v_july_v/article/details/6890054)。

大众点评前面还有 30 道逻辑题，15 道文字推理，15 道数学推理，一共只给 20min。

48. 9 月 28 日，网易笔试题：

1、英雄升级，从 0 级升到 1 级，概率 100%。

从 1 级升到 2 级, 有 1/3 的可能成功; 1/3 的可能停留原级; 1/3 的可能下降到 0 级;  
从 2 级升到 3 级, 有 1/9 的可能成功; 4/9 的可能停留原级; 4/9 的可能下降到 1 级。  
每次升级要花费一个宝石, 不管成功还是停留还是降级。

求英雄从 0 级升到 3 级平均花费的宝石数目。

点评: 题目的意思是, 从第  $n$  级升级到第  $n+1$  级成功的概率是  $(1/3)^n$  (指数), 停留原级和降级的概率一样, 都为  $[1-(1/3)^n]/2$ 。

2、将一个很长的字符串, 分割成一段一段的子字符串, 子字符串都是回文字符串。  
有回文字符串就输出最长的, 没有回文就输出一个一个的字符。

例如:

habbafgh

输出 h,abba,f,g,h。

点评: 编程艺术第十五章有这个回文问题的解答, 参见:

[http://blog.csdn.net/v\\_july\\_v/article/details/6712171](http://blog.csdn.net/v_july_v/article/details/6712171)。此外, 一般的人会想到用后缀数组来解决这个问题, 其余更多的方法请见: <http://dsqiu.iteye.com/blog/1688736>。

最后, 还可以看下这个链接:

<http://www.51nod.com/question/index.html#!questionId=672>。

#### 49. 10 月 9 日, 腾讯一面试题:

有一个 log 文件, 里面记录的格式为:

QQ 号:            时间:            flag:

如 123456        14: 00: 00        0

123457        14: 00: 01        1

其中 flag=0 表示登录 flag=1 表示退出

问: 统计一天平均在线的 QQ 数。

点评: 类似于此文中: <http://blog.csdn.net/hackbuteer1/article/details/7348968>, 第 8 题后的腾讯面试题, 读者可以参看之。

#### 50. 10 月 9 日, 腾讯面试题:

1. 有一亿个数, 输入一个数, 找出与它编辑距离在 3 以内的数, 比如输入 6(0110), 找出 0010 等数, 数是 32 位的。

2. 每个城市的 IP 段是固定的, 新来一个 IP, 找出它是哪个城市的, 设计一个后台系统。

#### 51. 10 月 9 日, YY 笔试题:

1 输出一个字符串中没有重复的字符。如 “baaca” 输出 “bac”。

2 对于一个多叉树, 设计 TreeNode 节点和函数, 返回先序遍历情况下的下一个节点。

函数定义为 `TreeNode* NextNode (TreeNode* node)`

3 分割字符串。

对于一个字符串，根据分隔符 separator，把字符串分割，如果存在多个分隔符连在一起，则当做一个分隔符。如果分隔符出现在“ ”符号之间，则不需要分割“ ”之间的字符。

比如 a++abc ，分隔符为+，输出 a abc

a+"hu+" 输出 a hu+

a++"HU+JI 输出 a "HU JI。

请根据上述需求完成函数: void spiltString(string aString, char aSeperator)。

## 52. 10月9日，赶集网笔试

赶集网最后4个程序/设计题

- 1.使用递归算法来判断数组a[N]是否为递增数组。
- 2.一个N\*N的矩阵，采用螺旋的方式打印矩阵中的数据。

```
1  2  3  4  5
6  7  8  9 10
11 12 13 14 15 16
17 18 19 20 21 22
```

输出顺序为1 2 3 4 5 10 16 22 21 20 19 18 17 6 7 8 9 15 14 13 12 11

- 3.两个数组有序，把两个数组中的元素合并到一个数组，合并后的数组同样有序，算法复杂度要求最低。

- 4.一个应用，系统中存有很多个人信息(身份证号，姓名)，请设计一个缓存算法，对于那些访问频率很大的个人信息，提高系统的性能。

## 53. 10月9日，阿里巴巴 2013 校园招聘全套笔试题（注：下图中所标答案不代表标准答案，有问题，欢迎留言评论）

第一部分 单选题 (每题 2 分, 共 20 分)

1. BMP 是 Windows 操作系统中的标准图像文件格式。24bit BMP 是真彩图, 记录了图片中各个点的 RGB 值。一张  $512 \times 1024$  点阵的图片, 其 BMP 真彩位图数据占 1.5M 字节。  
A. 0.5M B. 0.75M C. 1.5M D. 12M
2. 十进制数 10 的 3 进制 4 位数补码形式是 2122。  
A. 0101 B. 1010 C. 2121 D. 2122
3. 相对于传统磁盘, 固态硬盘——SSD 性能最大的优势是 随机读写速度快。  
A. 随机读写速度快 B. 连续读写速度快 C. 缩小磁盘尺寸 D. 提高磁盘寿命
4. 3 个进程共享 5 个等价的互斥资源, 进程每次申请资源都在其中任选一个申请。为使系统不死锁, 每个进程最多可以同时申请 2 个资源。  
A. 1 B. 2 C. 3 D. 4
5. 两个线程并发执行以下代码, 假设 a 是全局变量, 那么以下输出中 7 6 是不可能的。  

```

void foo()
{
    a = 5;
    a = a + 1;
    printf("%d", a);
}

```

A. 5 6 B. 6 5 C. 6 7 D. 7 6
6. 执行以下代码, 不可能的输出结果是 0 1。  

```

void main ()
{
    pid_t pid;
    pid = fork();
    if (pid < 0)
        printf("-1");
    else if (pid == 0)
        printf("0");
    else
        printf("1");
}

```

A. 0 B. -1 C. 0 1 D. 1 0
7. 在执行 广度优先搜索 操作时, 需要使用 队列 做辅助存储空间。  
A. 查找哈希 (Hash) 表 B. 深度优先搜索树  
C. 前序 (根) 遍历二叉树 D. 广度优先搜索树

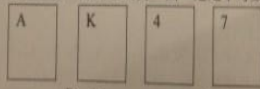
8. 关于排序算法的以下说法, 错误的是\_\_\_\_\_。
- A. 归并排序的平均时间复杂度为  $O(n \log n)$ , 最坏时间复杂度为  $O(n^2)$
  - B. 快速排序的平均时间复杂度为  $O(n \log n)$ , 最坏时间复杂度为  $O(n^2)$
  - C. 堆排序的平均时间复杂度为  $O(n \log n)$ , 最坏时间复杂度为  $O(n \log n)$
  - D. 冒泡排序的平均时间复杂度为  $O(n^2)$ , 最坏时间复杂度为  $O(n^2)$

9. 判断一包含  $n$  个整数的数组  $a[]$  中是否存在  $i, j, k$  满足  $a[i] + a[j] + a[k] = 0$  的时间复杂度需要\_\_\_\_\_。
- A.  $O(n^3)$
  - B.  $O(n^2 \log n)$
  - C.  $O(n^2)$
  - D.  $O(n \log n)$

10. 从 6 双不同颜色的皮鞋中任取 4 只, 其中恰好有一双同色的取法有\_\_\_\_\_种。
- A. 60
  - B. 120
  - C. 180
  - D. 240

## 第二部分 不定项选择题 (每题全选对得 5 分, 漏选得 2 分, 错选、不选得 0 分; 共 20 分)

11. 可以用来检测服务器上端口是否打开的命令是\_\_\_\_\_。
- A. ☒ ping
  - B. ☐ traceroute
  - C. ☐ telnet
  - D. ☐ netstat
12. 以下描述错误的有\_\_\_\_\_。
- A. 有些树 (Tree) 比哈希 (Hash) 快。
  - B. LCD 液晶显示全黑比显示全白更省电。
  - C. 32 位计算机可以管理超过 4G 内存。
  - D. ☒ main 函数不写 return 也能返回 0。
  - E. 低优先级 process 可以优先于高优先级 process 被 OS 调度运行
13. 正则表达式中 `\b` 是个特殊字符, 表示单词边界, 而不是空格。下列字符串中能匹配正则表达式 `\bhi\b.*\bABC\b` 的有\_\_\_\_\_。
- A. ☒ `@\bhi\bABC\b`
  - B. ☐ `@hi\bABC`
  - C. ☐ `@hi 来自北京的 ABC`
  - D. ☐ `@hi!ABC`
  - E. ☐ `@hiABC`
14. 有一副牌, 每张牌的正反面各印有 {2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A} 集合中的一个元素, 可重复。其中四张的一面如图所示。为了验证一个可能存在的规律: “如果某张扑克牌的一面是一个元音字母, 那么它的另一面一定是个奇数”, 必须翻开的牌有\_\_\_\_\_。



- A. A
- B. K
- C. 4
- D. 7



15. 二分查找又称折半查找。优点是比较次数少，查找速度快，平均性能好；其缺点是要求待查表为有序表，且插入删除困难。请先补全下面的 `binsearch` 代码，已知数字的范围为 `short` 可表示的范围，即 0 到  $2^{12}-1$ 。此题有两个空格，各 5 分)

```
short binsearch(short x, short A[], short n)
//x 为待找数字，A[] 为查找数字范围，n 为 A 的数组大小
//0<=n<=length(A); A 有序；查找成功返回结点下标，失败返回 -1
{
    short lower = 0;
    short upper = n;
    while (lower < upper)    //此处不计比较数
    {
        short mid = (lower + upper) / 2;
        if (A[mid] == x)    //此处计比较次数
            return mid;
        else if (A[mid] < x)    //此处计比较次数
            lower = mid + 1;
        else //A[mid] > x, 此处不计比较数
            upper = mid;
    }
    return -1; //当 lower>=upper 时表示查找区间为空，查找失败
}
```

并请回答：

在数组  $A[] = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23\}$  中查找元素 17，调用下面函数 `binsearch(17, A, 12)` 所花的  $A[mid]$  和  $x$  的值的比较次数是 2 次。

16. 某二叉树的前序遍历序列为  $-+a*b-cd/ef$ ，后序遍历序列为  $abcd-*+ef/-$ ，问其中序遍历序列是什么？

$abc-d-*+e/f$

$a+b*c-d-e/f$

17. 某虚拟内存系统采用 LRU 算法的页式内存管理，考虑下面页面访问地址流(每次访问在一个时间单位内完成)：

1, 8, 1, 7, 8, 2, 7, 2, 1, 8, 3, 8, 2, 1, 3, 1, 7, 1, 3, 7

假定内存容量为 4 个页面，开始是空的，则页面失效的次数为 8。

上述第 15 题，填空：  $lower + (upper - lower) / 2$

lower mid upper

0 6 12

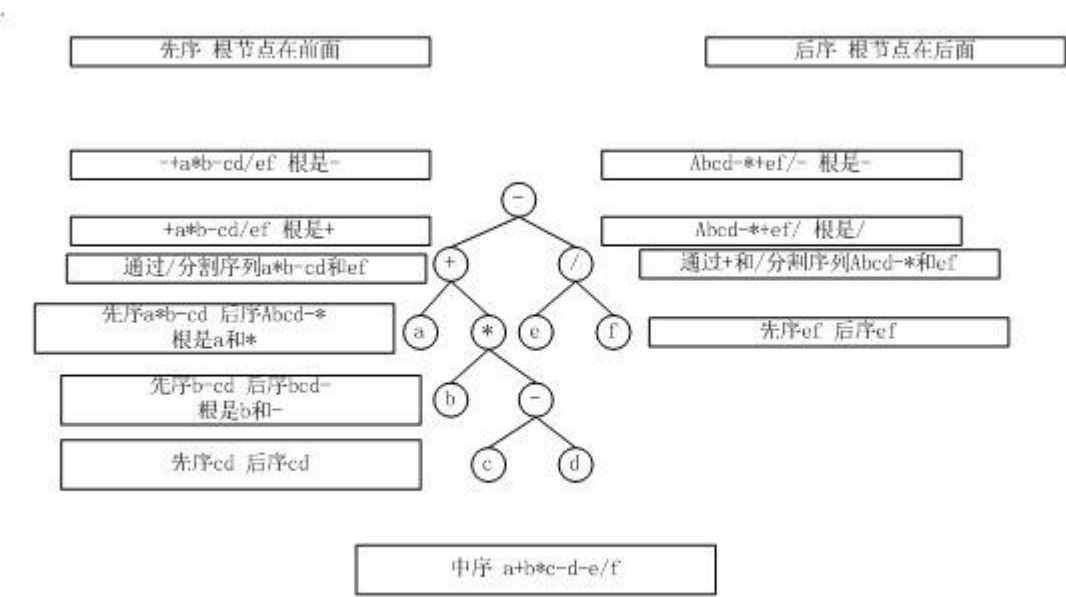
7 9 12

7 7 8

8 8 8

比较 4 次

上述第 16 题，解答如下图所示：



上述第 17 题，解答如下图所示：

访问	缓存1	缓存2	缓存3	缓存4	缺页中断
1	1				1
8	8	1			1
1	1	8			0
7	7	1	8		1
8	8	7	1		0
2	2	8	7	1	1
7	7	2	8	1	0
2	2	7	8	1	0
1	1	2	7	8	0
8	8	1	2	7	0
3	3	8	1	2	1
8	8	3	1	2	0
2	2	8	3	1	0
1	1	2	8	3	0
3	3	1	2	8	0
1	1	3	2	8	0
7	7	1	3	2	1
1	1	7	3	2	0
3	3	1	7	2	0
7	7	3	1	2	0
缺页次数					6

18、甲包 8 个红球 2 个蓝球，乙包 2 个红球 8 个蓝球。抛硬币决定从哪个包取球，取了 11 次，7 红 4 蓝。注，每次取后还放进去，只抛一次硬币。问选的是甲包的概率？

点评:

贝叶斯公式 + 全概率公式作答 (参看链接:

<http://www.doc88.com/p-132711202556.html>)。具体解答如下图所示:

设选甲包为事件 A, 选乙包为事件 B, 取了 11 次, 7 红 4 蓝为事件 C, 那么有

$$p(A) = 1/2$$

$$p(B) = 1/2$$

甲包中发生 C 事件的概率为  $p(C|A) = C_{11}^4 0.8^7 0.2^4$  (二项分布)

乙包中发生 C 事件的概率为  $p(C|B) = C_{11}^4 0.8^4 0.2^7$  (二项分布)

当 C 发生时 A 发生的概率为

$$p(A|C) = \frac{p(C|A)p(A)}{p(C)} = \frac{p(C|A)p(A)}{p(C)} = \frac{p(C|A)p(A)}{p(A)p(C|A) + p(B)p(C|B)} = \frac{0.512}{0.52} \approx 0.98 \quad (\text{贝叶斯公式 全概率公式})$$

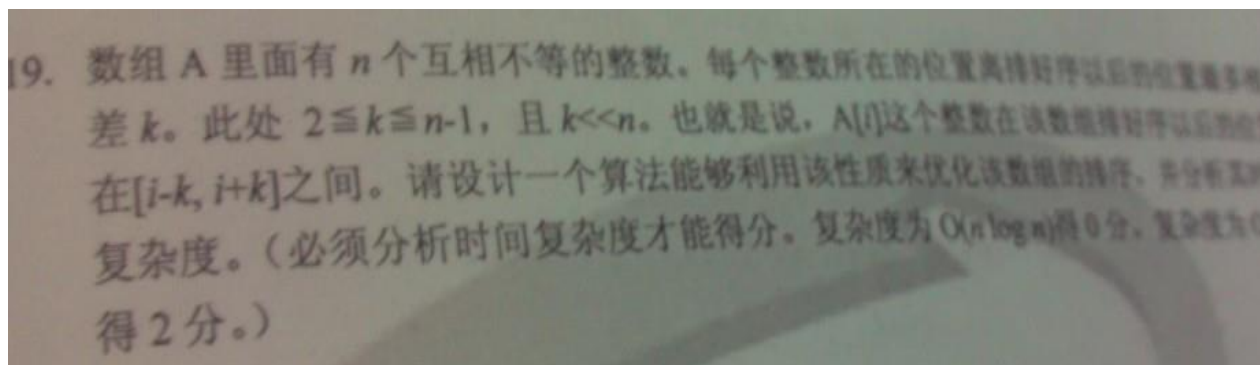
注: 上述第 15~18 的解答全部来自读者 Lei Lei 来信给出的解答, 他的博客地址是:

<http://blog.csdn.net/nwpulei>, 特此感谢。有任何问题, 欢迎随时讨论&指正, 同时, 更欢迎其他朋友也一起来做这些题目 (你的答案一经选用, 我可以根据你的要求, 贴出你的个人主页或微博地址或博客地址)。

19、已知一个  $n$  个元素的数组, 第  $i$  个元素在排序后的位置在  $[i-k, i+k]$  区间,  $k < n$ 。

让你设计一个算法对数组排序, 要求时间复杂度最小,  $O(n \log n)$  不得分,  $O(nk)$

得 2 分, 如下图所示:



#### 第四部分 综合题 (20 分)

20. 有一个序列  $seq=[a,b,c,d,aa,ba,ca,da,ab,bb,cb,db,ac,bc,cc,dc,\dots,aaa,baa,caa,\dots]$ , 其中第 1 个字符串是 a。请问(1) aaa 是 seq 中第几个字符串? (2) ababacd 是 seq 中第几个字符串? (3) seq 中第 1000 个字符串是什么? (4) 编程实现 find 函数“输入一个子串, 返回所在位置”, 编程语言不限。

读者 twtsa 毛遂自荐, 这是他给出的上述第 19~20 题的个人题解:

<http://blog.csdn.net/twtsa/article/details/8055143>。有任何问题, 欢迎随时讨论&指正。

54. 10 月 10 日, 暴风影音笔试:

都是非常基础的题目, 这是其中一道: 一个整数转换成二进制后, 问里面有多少个 1。

55. 10 月 10 日, 2013 亚马逊在线笔试题目

题目及参考答案请见这: <http://blog.chinaunix.net/uid-26750075-id-3370694.html>。(感谢读者 freeloki 来信提供)。

56. 10 月 10 日人人网面试题

第一面:

1、(1) ++i 和 i++, 那个效率高?

(2) ++++i, i++++, 哪个是合法的?

(3) 实现 int 型的 ++i 和 i++ 操作。

2、一段程序, 求输出。(考察静态变量和模版类)

```
1. int g = 0;
2. template<typename T>
3. class B
4. {
5. public:
6.     int static fun()
7.     {
8.         static int value = ++g;
9.         return value;
10.    }
11. };
12.
13. int main()
14. {
15.     cout << B<int>::fun() << endl;
```

```
16.     cout << B<char>::fun() << endl;
17.     cout << B<float>::fun() << endl;
18.     cout << B<int>::fun() << endl;
19.     cout << B<long>::fun() << endl;
20.     return 0;
21. }
```

3、（1）实现二进制转十进制。

（2）如果有下面这种能直接求二进制转十进制的代码，是怎么实现的？

```
binary<1>::value; // 结果为 1
```

```
binary<11>::value; // 结果为 3
```

4、volatile、explicit、mutable 表示的含义。

5、求整形数组的一个子数组，使得该子数组所有元素的和的绝对值最大。

6、（1）写求单链表是否有环的算法。

（2）如果有环，如何找出环的第一个结点。

7、实现单例模式。

二面：

1、一个文本，一万行，每行一个词，统计出现频率最高的前 10 个词（词的平均长度为 Len）。并分析时间复杂度。

2、求数组中最长递增子序列。

57. 10月10日，网易2013校园招聘全套笔试题：

网易 NetEase  
www.163.com

网易 2013 校园招聘

## 网易 2013 校园招聘笔试题

2

【适用职位：C++开发，客户端安全开发，客户端安全开发，移动平台开发(机研&邮件)】

座位号：\_\_\_\_\_

申请职位：\_\_\_\_\_

姓名：\_\_\_\_\_

联系电话：\_\_\_\_\_

### 笔试说明：

1. 所有题目请在答题纸上作答。
2. 笔试时间为 2 小时。
3. 笔试结束时请务必将所有答题纸、草稿纸、试卷全部交回。

### 第一部分（必做）：计算机基础类（25 分）

（所有的选择题都是多项选择）

1. (2 分) 假设进栈次序是 e1,e2,e3,e4,则可能的出栈次序可能是：( )  
A e2,e4,e3,e1 B e2,e3,e1,e1 C e3,e2,e4,e1 D e1,e2,e4,e3

2. (2 分) 表达式“X=A+B\*(C-D)/E”的后缀表示形式可以是：( )  
A. XAB+CDE/\*=  
C. XABCD-\*E/+=  
B. XA+BC-DE/\*=  
D. XABCDE+\*/=

3. (2 分) 以下排序算法是非稳定排序的是 ( )  
A. 冒泡排序 B. 归并排序 C. 快速排序 D. 堆排序 E 希尔排序

4. (2 分) 一个包含 n 个节点的四叉树，每个节点都有四个指向孩子节点的指针，这 4n 个指针中有 ( ) 个空指针。

5. (2 分) int func(unsigned i)

```
{
    unsigned temp = i;
    temp = (temp & 0x55555555) + ((temp & 0xaaaaaaaa) >> 1);
    temp = (temp & 0x33333333) + ((temp & 0xcccccccc) >> 2);
    temp = (temp & 0xf0f0f0f0) + ((temp & 0xf0f0f0f0) >> 4);
    temp = (temp & 0x00ff) + ((temp & 0xff00ff) >> 8);
    temp = (temp & 0xffff) + ((temp & 0xffff0000) >> 16);
    return temp;
}
```

}. 请问 func(0x7530829) 的返回值是 ( )

A) 15 B) 16 C) 17 D) 18

1/7

试卷编号：02 卷\_C++开发

网易 NetEase  
www.163.com

网易 2013 校园招聘

6. (2分)进程与线程的差别有 ( )

A. 操作系统只调度进程, 不调度线程  
B. 线程共享内存地址空间, 进程不共享  
C. 线程间可以共享内存数据, 但进程不可以  
D. 线程间可以通过 IPC 通信, 但线程不可以

7. (2分)段页式管理中, 地址映像表是 ( )

A. 每个进程一张段表, 一张页表  
B. 进程的每个段一张段表, 一张页表  
C. 每个进程一张段表, 每个段一张页表  
D. 每个进程一张页表, 每个段一张段表

8. (2分)关于 TCP 协议, 下面哪种说法是错误的 ( )

A. TCP 关闭过程中两端的 socket 都会经过 TIME\_WAIT 状态  
B. 对一个 Established 状态的 TCP 连接, 调用 shutdown 函数可以让主动调用的一方进入半关闭状态  
C. TCP 协议默认保证了当 TCP 的一端发生意外崩溃(当机/网线断开或路由故障), 另一端能自动检测到连接失效。  
D. 在成功建立连接的 TCP 连接上, 只有在 Established 状态才能收发数据, 其他状态都不可以。

9. (2分)关于主键(primary key)和索引(index)的说法哪些是错误的? ( )

A) 唯一索引的允许为 NULL 值  
B) 一个关系表中的主键必定是另一表中的主键。  
C) 一个表只能有一个唯一性索引  
D) 索引主要影响查询流程, 对数据的插入影响不大。

10. (2分)数据库的事务隔离级别一般分为 4 个级别, 其中可能发生“不可重复读”的事务级别有 ( )

A)SERIALIZABLE B)READ COMMITTED C)READ UNCOMMITTED D)REPEATABLE READ

11. (5分) 如果设  $F(n)$  为该数列的第  $n$  项 ( $n \in \mathbb{N}^+$ )。那么这句话可以写成如下形式:

$$F(1)=1,$$

$$F(2)=1,$$

$$F(n)=F(n-1)+F(n-2) \quad (n \geq 3).$$

请实现该函数  $F(n)$  的求解, 并给出算法复杂度, 要求算法复杂度小于  $O(n^2)$ 。

## 第二部分 (必做): 程序设计 (25 分)

1. (2分) 下面的程序的输出是什么?

```
#include <stdio.h>
int main()
{
    int n;
    char y[10] = "nise";
```

2 / 7

试卷编号: 02\_卷\_C++开发

```
char *x = y;
n = strlen(x);
x = x[n];
x++;
printf("x=%s\n", x);
printf("y=%s\n", y);
}
```

2. (2 分) 请给出下面程序的输出结果, 并说明原因

```
#include <vector>
#include <iostream>

using namespace std;

template<class T>
class array {
public:
    array(int size):
        size_t getVectorSize() { return _data.size(); }
        size_t getSize() { return _size; }

public:
    vector<T> _data;
    size_t _size;
};

template<class T>
array<T>::array(int size)
: _size(size), _data(_size)
{}

int main()
{
    array<int> *arr = new array<int>(3);
    cout << arr->getVectorSize() << endl;
    cout << arr->getSize() << endl;
    return 0;
}
```

3 / 7

试卷编号: 02\_卷\_C++开发

3. (2 分) CAS (CompareAndSwap), 是用来实现 lock-free 编程的重要手段之一, 多数

处理器都支持这一原子操作, 其伪代码描述如下,

```
template bool CAS(T* addr, T expected, T value)
{
    if (*addr == expected) {
        *addr = value;
        return true;
    }
    return false;
}
```

请完成下面填空, 实现全局计数器的原子递增操作

```
int count = 0;
void count_atomic_inc(int *addr)
{
    int oldval = 0;
    int newval = 0;

    do{
        oldval = *addr;
        newval = ____ + 1;
    } until CAS(____, ____, ____);
}
```

4. (2 分) 下面的程序会输出几个“\_”?

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(void)
{
    int i;
    for(i=0; i<2; i++){
        fork();
        printf("_");
        fflush(stdout);
    }
    return 0;
}
```

5. (4 分) 写程序判断当前 CPU 是大端 CPU 还是小端 CPU, 并做简要说明

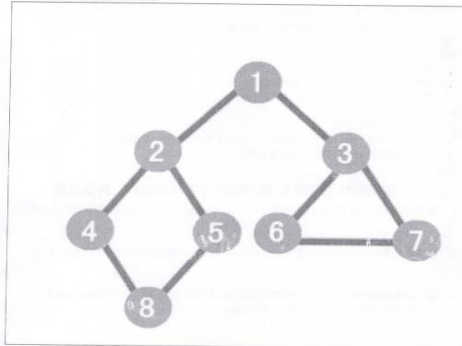
4 / 7

试卷编号: 02\_卷\_C++开发



6. (5 分) 利用位运算实现两个整数的加法运算, 请代码实现, 并做简要说明

7. (8 分) 图深度遍历问题



- 写出上述图的深度优先遍历的顺序 (遍历起点是节点 1)
- 若用邻接矩阵 Matrix 存储该图, 写出该矩阵
- 若用非递归方式实现深度优先遍历, 请叙述大致的实现思想?
- 用你熟悉的任何语言实现非递归深度优先遍历

### 第三部分 (选做): C++ 开发工程师必做, 其他选做 (15 分)

1. (6 分) 给定一个巨大的文本文件, 写一个程序随机输出文件任意 k 行 (k 不大, k 行能放入内存), 要求每一行出现概率相等, 请给出核心算法、算法复杂度以及简要的算法原理说明。

2. (9 分) Spin Lock 是一种较为常见与使用的互斥方法, 下面是一种其实现方式:

```
typedef int lock_t;

void initlock(volatile lock_t* lock_status){
```

5 / 7

试卷编号: 02 卷\_C++ 开发

```
*lock_status = 0;
}

void lock(volatile lock_t* lock_status) {
    while(test_and_set(lock_status) == 1)
        ;
}
```

```
void unlock(volatile lock_t* lock_status) {
    *lock_status = 0;
}
```

- volatile 关键字的作用
- 怎样优化 lock 函数 (提示: 多 CPU 下如何提高 CPU Cache 效率)
- 上述代码可能存在的问题 (内存模型考虑)

### 第四部分 (选做): 客户端安全工程师必做, 其他选做

1. 请简述使用互斥量 (Mutex) 和临界区 (CriticalSection) 作为同步方法的区别及应用场景。

2. 如何让一个循环执行的线程安全退出, 请用 C++ 代码实现相应线程函数及退出机制。

3. 请列举可以用来唯一标识一台机器特征的特性有哪些; 在某些机器特征可能被篡改情况下, 请设计一套方案用来唯一标识这台机器。

4. 请描述利用 CreateProcess 注入一个 dll 到新创建进程的过程, 必要部分可以写伪代码。

5. 请列举 32 位系统下, 驱动隐藏进程的常用方法。

6. 如何检测当前程序运行在虚拟机环境下。

说明: 虚拟机主要是指目前常见的 VMWare, Virtual PC, VirtualBox 等; 可以对一种虚拟机检测, 也可以对多种虚拟机检测进行说明。

### 第五部分 (选做): 客户端开发工程师必做, 其他选做

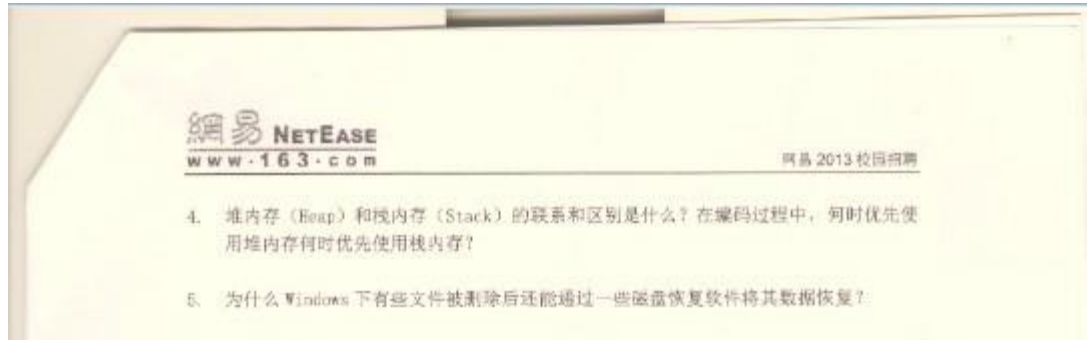
1. WM\_QUIT 消息的用途是什么? 一个普通的 Windows 窗口能收到的最后一条消息是什么?

2. 什么是 DLL 的延迟加载 (Delay Load)? 用延迟加载有什么好处?

3. 并行计算 (Parallel Computing) 与并发计算 (Concurrent Computing) 的联系与区别是什么?

6 / 7

试卷编号: 02 卷\_C++ 开发



58. 10月10日，网易，数据挖掘工程师：

- 1，简述你对数据与处理的认识；
- 2，简述你对中文分词的理解，说明主要难点和常用算法；
- 3，常见的分类算法有哪些；
- 4，简述 K-MEANS 算法；
- 5，设计一个智能的商品推荐系统；
- 6，简述你对观点挖掘的认识。

点评：其它题目与上述第 56 题第一部分

（<http://blog.csdn.net/hackbuteer1/article/details/8060917>）所述相同。

59. 10月11日，阿里巴巴笔试部分题目：

1. 甲乙两个人上街，捡到一张 10 块钱的购物卡，两人就想出一个办法来分配这张卡。两个分别将自己出的价格写在纸上，然后看谁出的价高就给谁，并且那个出价高的人要把出的钱给对方。现在甲有 6 块钱，乙有 8 块钱。问谁获得的钱多。（多选）

A 甲多      B 乙多      C 一样多      D 有可能出现有人赔钱的情况

2. 有一个怪物流落到一个荒岛上，荒岛上有  $n$  条鳄鱼。每条鳄鱼都有实力单独吃掉怪物。但是吃掉怪物是有风险的，会造成体力值下降，然后会有可能被掉其他鳄鱼吃。问，最后那个怪物是危险的还是安全的？

3. 算法题：

$A[i]$  是一个有序递增数组，其中所有的数字都不相等，请设计一种算法，求出其中所有的  $A[i]=i$  的数字并分析时间复杂度，不分析复杂度不得分。

4. 大题

你在浏览器中输入网址：[http://blog.csdn.net/v\\_JULY\\_v](http://blog.csdn.net/v_JULY_v)，按下回车键后，会发生什么事情，请一一描述（20 分）。包括浏览器，网络，服务器等等发生的事情，及各项关键技术。

点评：这样的题考过很多次，参考答案如下图所示：

回答：

简单来说有以下步骤：

- 1、查找域名对应的IP地址。这一步会依次查找浏览器缓存，系统缓存，路由器缓存，ISP DNS缓存，根。
- 2、向IP对应的服务器发送请求。
- 3、服务器响应请求，发回网页内容。
- 4、浏览器解析网页内容。

当然，由于网页可能有重定向，或者嵌入了图片，AJAX，其它子网页等等，这4个步骤可能反复进行多次才能展示给用户。

60. 10月11日，华为一面：

- 1、将一个普通的二叉树转换为二叉排序树？
- 2、随便写一个排序算法。

61. 10月11日，完美笔试题：

- 1.为什么析构函数应该设为虚函数
- 2.大数字乘法问题
- 3.双向链表模拟队列操作 push pop find
- 4.求  $a/3$  不能用除法
- 5.多核下多线程同步问题，使用锁应该注意什么
- 6.三个宝箱有一个里面有珠宝，现在拿第一个宝箱，然后打开第二个宝箱后发现没有珠宝，用概率论原理解释为什么现在拿第三个宝箱，里面有珠宝的概率比拿第一个宝箱高。

62. 10月11日，搜狐畅游旗下第七大道笔试题：

算法题

- 1.一个数是否是另一个数的平方。
- 2.N 进制换成 M 进制
- 3.设计一个大数乘法

综合题

- 1.N 个数，出栈有几种情况
- 2.进程死锁原因及条件.

63. 腾讯一个非常有意思的面试题：

N 个数组，每个数组中的元素都是递增的顺序，现在要找出这 N 个数组中的公共元素部分，如何做？注：不能用额外辅助空间。

点评：

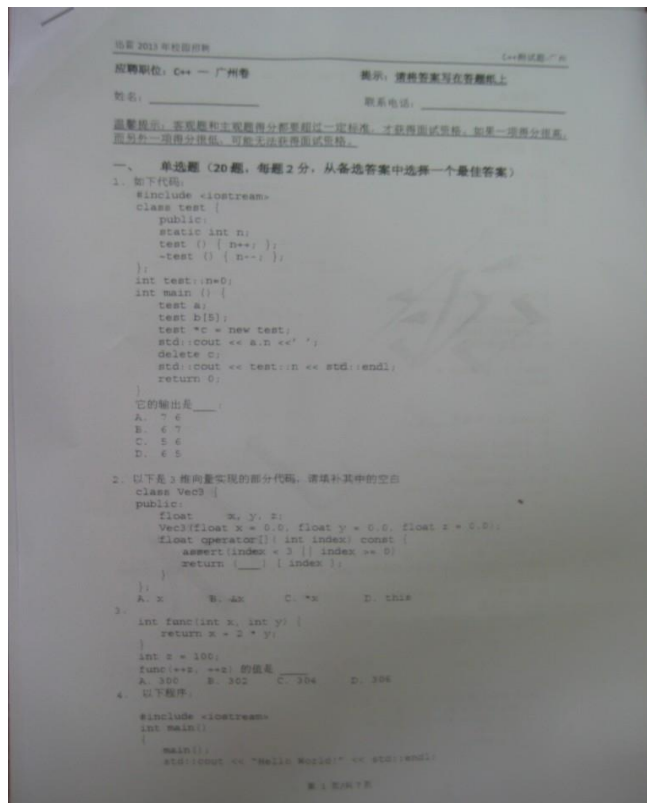
讨论了半天：<http://weibo.com/1580904460/z08mT0aFj>，没个好的结果，发现还是上午想到的 N 个指针逐步向后移动，辅以二分，然后 N 路归并更靠谱，类似这里的第

5 题所述的办法:

<http://www.cnblogs.com/BeyondAnyTime/archive/2012/07/17/2593224.html>。若读者有更好的思路, 欢迎赐教。

#### 64. 10 月 12 日, 迅雷 2013 校园招聘「广州站」C++方向全套笔试题

(注: 若照片看不清楚, 请右键点击“图片另存为”到桌面, 然后再打开图片, 便可以随意放大缩小图片拉)



6. 以下代码的输出是
- ```

#include <iostream>
using namespace std;
int main() {
    cout << "Hello World! Hello World! ...";
    return 0;
}

```
- A. Hello World!  
B. Hello World! Hello World! Hello World! ...  
C. 程序进入死循环, 没有输出  
D. 程序崩溃
7. 以下代码的输出是
- ```

int x = 12345; x = x << 31; x = x >> 31; x 的值是

```
- A. 0  
B. 1  
C. 12345  
D. 54321
8. 以下代码的输出是
- ```

class X {
public:
    virtual ~X(){};
};
sizeof(X) 的值是

```
- A. 0  
B. sizeof(int)  
C. sizeof(long)  
D. 以上都有可能
9. 以下代码的输出是
- ```

class Y: public X {
public:
    ~Y(){};
};
sizeof(Y) 的值是

```
- A. 0  
B. sizeof(int)  
C. sizeof(long)  
D. 以上都有可能
10. 以下代码段的输出是
- ```

int x = 7;
if (x == 7/2*2) {
    cout << "a";
    cout << "b";
    cout << "c";
}

```
- A. abc B. bc C. c D. d
11. 对数组一个大小为 16 的随机整数数组 int a[16] 按升序排序, 适合采用以下的哪种排序算法?
- A. 插入排序 B. 归并排序 C. 快速排序 D. 希尔排序
12. 以下三个函数
- ```

void swap_gtl(int& a, int& b) {
    if (a > b) {
        int tmp = a;
        a = b;
        b = tmp;
    }
}

void swap_gtr(int& a, int& b) {
    int tmp = a + b;
    a = tmp - b;
    b = tmp - a;
}

void swap_gtl2(int& a, int& b) {
    int tmp = a + b;
    a = tmp - b;
    b = tmp - a;
}

```

13. 以下哪个运算符具有最高的优先级?
- A. \* B. + C. [] D. &
14. 以下代码段的输出是
- ```

int i = 2;
double j = 2.0;
if (i == j) {
    cout << "OK.";
}
else {
    cout << "Not OK.";
}

```
- A. OK Not OK.  
B. Not OK.  
C. OK.  
D. 以上都不对
15. 如果一个类中含有纯虚函数, 则该类别称为
- A. 基类 B. 派生类 C. 抽象类 D. 虚基类
16. 有以下定义, 以下说法中正确的是
- ```

typedef union {
    unsigned int u;
    int i;
    char c;
    std::string s;
} u;

```
- A. sizeof(u) = sizeof(int)  
B. sizeof(u) = sizeof(std::string)

- C. 取决于字符串的长度  
D. 编译错误
17. 对任意给定的 5 个数进行排序，在最坏情况下只需要\_\_\_\_\_次比较  
A. 5 B. 6 C. 7 D. 8
18. 以下代码的输出是  
#include <iostream>  
struct MyBase  
{  
int x;  
template <int RANGE>  
virtual void print() {  
std::cout << "RANGE = " << RANGE << std::endl;  
}  
}  
struct MyDerived : public MyBase  
{  
template <int RANGE>  
void print() {  
std::cout << "RANGE = " << RANGE << std::endl;  
}  
}  
int main(int argc, char\*\* argv)  
{  
MyBase\* b = new MyDerived;  
b->x = 1;  
b->print<5>();  
return 0;  
}
- A. 2 B. 7 C. 8 D. 编译错误
19. 单件模式一般通过\_\_\_\_\_来实现  
A. static B. public C. auto D. volatile
20. 在类 ClassA 的定义及其成员函数定义的外部，可以被访问的成员有  
A. 所有成员  
B. private 或 protected 成员  
C. public 成员  
D. public 或 protected 成员
- 二、多选题 (10 题，每题 3 分，漏选或错选都不得分)
1. 设计模式中的模板方法模式可以利用 C++ 的哪些语言特性来实现?  
A. 虚继承 B. 虚函数 C. 继承 D. 静态函数
2. 如下代码:  
union {  
struct {  
unsigned char a[4];  
unsigned char b[4];  
};  
unsigned char c;  
} u;  
u.a = 0xFF;  
u.b = 0xFF;  
则 u.c 的值可能是  
A. 0xFF B. 0xFF0 C. 240 D. 15
3. 以下哪些是 C++ 的关键字  
A. mutable B. auto C. register D. volatile
4. C++ 支持以下哪些编程风格  
A. 泛型 B. 面向过程 C. 基于对象 D. 面向对象

5. 以下算法中，最坏情况下的时间复杂度是  $O(n^2)$  的有  
A. 快速排序 B. 插入排序 C. 希尔排序 D. 冒泡排序
6. 下列类的定义式中，合法的有  
A. template <typename T>  
class ClassA {  
class ClassB : public ClassA {  
};  
};  
B. class ClassB : public ClassA {  
};  
C. class ClassC : virtual ClassB {  
};  
D. class ClassD : public ClassC, public ClassB {  
};
7. 以下的数据结构中，插入操作的时间复杂度是  $O(\log n)$  的有  
A. 红黑树 B. 二叉树 C. B 树 D. AVL 树
8. C++ 中不能被派生类继承的函数是  
A. 虚函数 B. 构造函数 C. 静态成员函数 D. 析构函数
9. 以下代码片段  
class A {  
public:  
A() {}  
A(const std::string &s) : str(s) {}  
std::string str;  
};  
int main() {  
A a;  
a = std::string("haha");  
return 0;  
}
- 代码 a = std::string("haha")，调用了哪些函数?  
A. A::A(const std::string &)  
B. A::A()  
C. A::operator=(const A &)  
D. A::A()
10. 关于友员的概念，以下说法正确的是  
A. 友员函数没有 this 指针  
B. 调用友员函数时必须在其的实例中给出访问的对象  
C. 一个类的成员函数也可以作为一个类的友员函数  
D. 只能在类的公有域声明友员
- 三、填空题 (15 空，每空 2 分)
1. 以下代码利用欧几里德算法计算两个整数的最大公因子，欧几里德算法依赖于以下函数  
gcd(a,b) = gcd(b, a mod b), gcd(a,0) = a, gcd(0,b) = b  
请填写其中的空白:  
class Math  
{  
public:  
int gcd(int a, int b) {  
while (true) {  
if (a == 0) {  
return b;  
}  
if (b == 0) {  
return a;  
}  
a = a % b;  
b = b % a;  
}  
}

2. 以下是一个有理数类的一个实现，实现了有理数的乘法(\*), 相反数(-), 化简(simplify)操作，其中用到了填空题；中实现的 Math::gcd() 函数，请填写其中的空白

```
class Rational
{
public:
    Rational(int n = 0, int d = 1) : numer_(n), denom_(d) {}
    Rational(int n, int d) : numer_(n), denom_(d) {
        simplify();
    }
    ~Rational() {}
    int numerator() const { return numer_; }
    int denominator() const { return denom_; }
    /* operator*() signature */
    {
        (4)
        int gcd1 = Math::gcd(numer_, x.denom_);
        int gcd2 = (5);
        numer_ = (numer_/gcd1) * (x.numer_/gcd2);
        denom_ = (6);
    }
    /* operator+=() signature */
    {
        (8)
        return (numer_ += x.numer_, (9) (denom_ += x.denom_));
    }

    void simplify() {
        (10) (denom_ != 0);
        if (numer_ == 0) {
            (11);
            return;
        }
        int g = (12);
        numer_ /= g;
        denom_ /= g;

        if ( (13) ) {
            numer_ = -numer_;
            denom_ = -denom_;
        }
    }

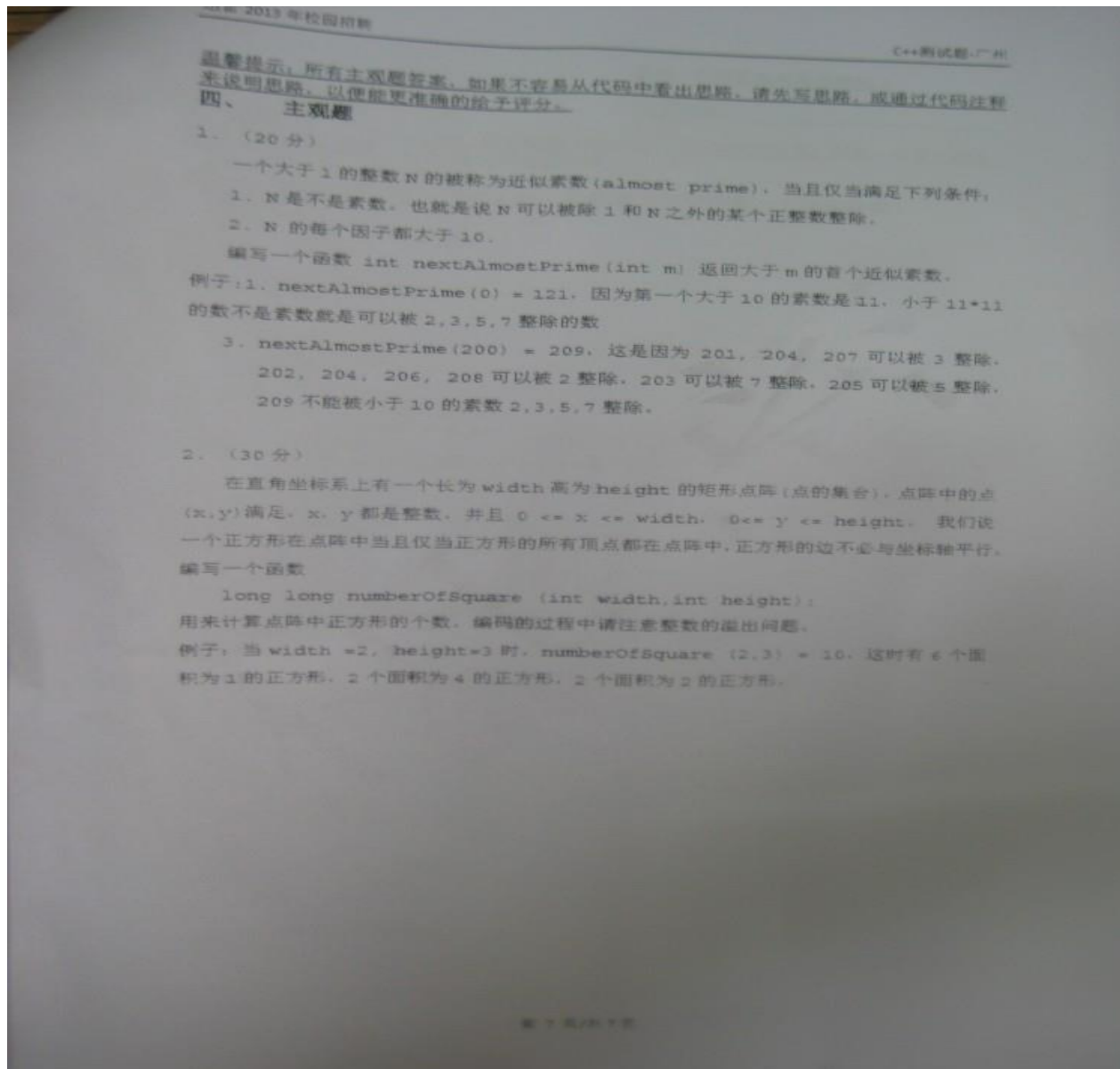
private:
    int numer_; // numerator
    int denom_; // denominator
};

inline Rational operator* (const Rational& r)
{
    return (14);
}

inline Rational operator+ (const Rational& a, const Rational& b)
{
    Rational r(a);
    (15);
    return r;
}

```





65. 10月12日晚，微策略北京站笔试题（根据读者回忆整理）：

1、魔术定义：整数  $N$  以基数  $B$  表示，如 21 以基数 3 表示为 210，那么 21 是基数 3 的一个魔术，210 三个位的值都不一样。设计函数，输入参数  $N$  和  $B$  ( $B$  介于 2 到 10 之间)，返回是否为魔术。

2、斐波那契数列的变形，一个贼每次上楼梯 1 或者 2，一个 27 层的楼梯需要多少种方法，记住贼不能经过 5,8,13 层，否则会被抓住。点评：还是可以用斐波那契来推算， $f(n) = f(n-1) + f(n-2)$ ，只是  $f(5) f(8) f(13) = 0$ ，

<http://www.51nod.com/answer/index.html#answerId=596>。

3、给定一棵树根节点，每个节点里面的值都不相同，查找 iKEY 的节点，并使用一个给定的节点将查找到的节点替换掉。节点内有两个孩子节点和一个父节点。

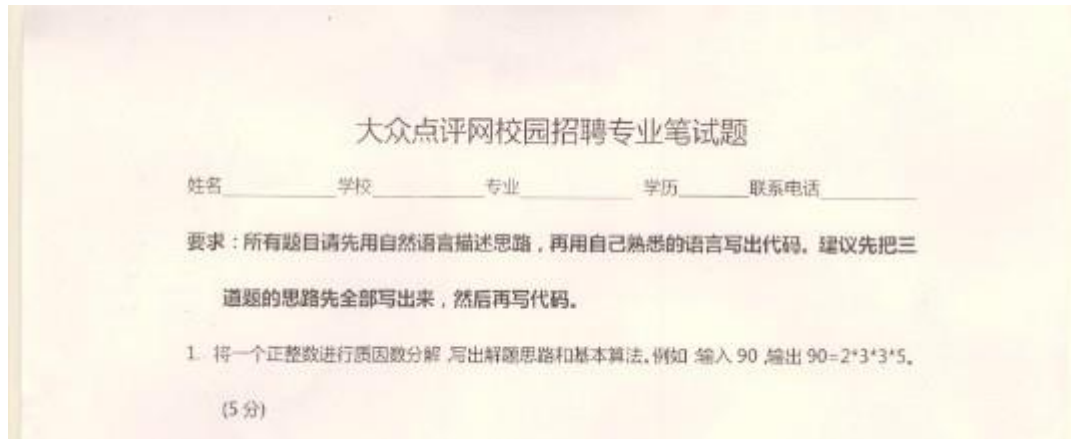
4、字符串数组  $S$ ，全是 0 和 1 表示的，字符串都是  $n$  位的，且 1 的个数小于等于  $l$ ，返回 index 的字符串。（这个比较奇怪，如果  $S$  中字符串都是符合 1 的个数小于等于  $l$ ，则直接

可以得到 `index` 的字符串啊，难道是要先求这个字符串数组？那就比较麻烦了）

5、降序排列的数组，找到其中两个不同的值，其乘积最接近一个给定的值  $M$ ，感觉和加法求和很类似。

6、序列  $123\dots N$ ， $N$  介于 3 和 9 之间，在其中加入  $+$  或者空格，使其和为 0，如  $123456\ 1-2\ 3-4\ 5+6\ 7$  等价于  $1-23-45+67=0$ 。请问，如何获得所有组合？

66. 10 月 12 日，大众点评笔试一题：

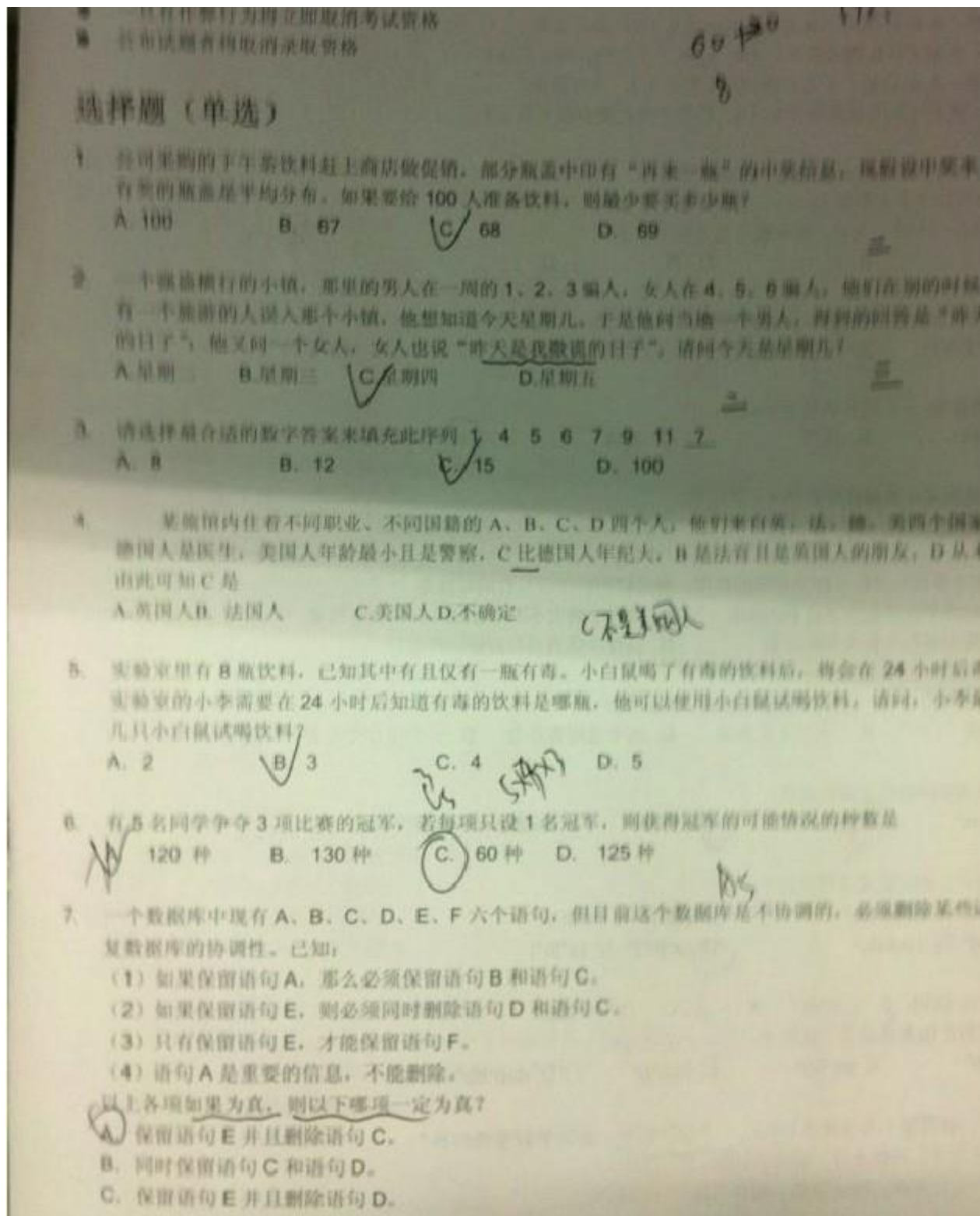


67. 读者私信，昨日（12 号）美团的笔试题：

- 1、一副扑克 52 张（去了大小王），洗牌，求最顶一张和最底一张是 A 的概率
- 2、知道两个数的异或以及这两个数的和，问可以确定这对数吗？为什么？给出推理过程
- 3、A、B 两个文件各存 50 亿个商品名称，每个 50 个字符，求这两个文件中相同名称的商品名，内存限制 4G（看过您的《教你如何迅速秒杀掉：99%的海量数据处理面试题》中的第 6 题，无压力，非常感谢）
- 4、给一个二叉树的后序遍历和中序遍历，画出这颗二叉树，写出前序遍历结果，并给出推理过程
- 5、一个有序数组 `array`，给一个数  $x$ ，可重复，求这个数在 `array` 中出现的区间，算法思路和代码实现
- 6、一个映射文件中存了 ip 地址区间和城市名称，形如：  
10.0.0.1 10.0.1.27 北京  
10.0.2.1 10.0.2.27 北京  
201.0.1.12 201.0.2.124 上海

给你一个 ip 地址，获取城市名称，要求：1）给出算法思想 2）代码实现。

68. 10月12日晚, 360 2013 校招部分笔试题(注: 图中标答案不代表正确答案):



```
int main()
{
fork()||fork();
```

```
return 0;
}
```

问，上述程序创建了几个进程？

```
bash-4.1$ ps
  PID   PPID  PGID   WINPID  TTY      UID   STIME  COMMAND
  7020    5488  6936    7020   cons0    1000  21:50:06 /cygdrive/g/lihuib
/mass/test/a
  4044    1052  4044    6716   cons1    1000  21:50:14 /usr/bin/ps
  5644      1  5644    5644   cons0    1000  11:35:57 /usr/bin/bash
  6936   5644  6936    3204   cons0    1000  21:50:06 /cygdrive/g/lihuib
/mass/test/a
  1052      1  1052    1052   cons1    1000  21:50:12 /usr/bin/bash
  5488   6936  6936    5488   cons0    1000  21:50:06 /cygdrive/g/lihuib
/mass/test/a
bash-4.1$
```

编程题、传教士人数  $m$ ，野人  $c$ ， $m \geq c$ ，开始都在岸左边，

①船只能载两人，传教士和野人都会划船，当然必须有人划船

②两岸边保证野人人数不能大于传教士人数

把所有人都送过河，设计一方案，要求编程实现。

点评：

读者 [huangxy10](#) 于本文评论下第 169 楼提供了一种解法：

<http://blog.csdn.net/huangxy10/article/details/8066408>。再附一个讨论帖子：

<http://topic.csdn.net/u/20121012/22/70226713-A669-4F03-80B7-BFFF12A330EB.html>。

69. 10 月 13 日，百度 2013 校招北京站笔试题：

一、简答题（30 分）

1、用简单语句描述数据库操作的步骤

2、写出 TCP/IP 的四层结构

3、什么是 MVC 结构，并描述各层结构的作用

二、算法与程序设计题（40 分）

1、字母  $a-z$ ，数字  $0-9$ ，现需要其中任意 3 个作为密码，请输出所有可能组合。（伪码\C\C++\JAVA）（10 分）

点评：如本文评论下第 198 楼所述，即从  $26+10=36$  个不同字符中选取 3 个字符的组合，用递归及非递归两种方法，可以参照以下链接：

<http://blog.csdn.net/wumuzi520/article/details/8087501>（从  $n$  个数中选取  $m$  个数的组合数），主要代码如下：

```
1. //copyright @wumuzi520
2. //从 n 个数中选取 m 个数的组合数
3. void Combination(int arr[], int nLen, int m, int out[], int outLen)
4. {
```

```

5.     if(m == 0)
6.     {
7.         for (int j = 0; j < outLen; j++)
8.         {
9.             cout << out[j] << "\t";
10.        }
11.        cout << endl;
12.
13.        return;
14.    }
15.
16.    for (int i = nLen; i >= m; --i) //从后往前依次选定一个
17.    {
18.        out[m-1] = arr[i-1]; //选定一个后
19.        Combination(arr,i-1,m-1,out,outLen); // 从前 i-1 个里面选取 m-1 个进行递
    归
20.    }
21. }
22.
23. void PrintCombination(int arr[], int nLen, int m)
24. {
25.     int* out = new int[m];
26.     Combination(arr,nLen,m,out,m);
27.     delete [] out;
28. }

```

2、实现字符串反转函数（10 分）

3、给定字符串函数 a、插入 b、删除 c、替换

例如字符串 A=acegf，字符串 B=adef，最少需要 2 步操作将 A 转换为 B，

即第一步将 c 替换为 d，第二步将 g 删除；

（1）请问将字符串 A=gumbo 转换为字符串 B=gambol，最少需要几步操作，列出如何操作（2 分）

（2）任意字符串 A 和字符串 B，如何计算最小操作次数，计算思路，并给出递归公式（3 分）

（3）实现代码（注意代码风格与效率）(15 分)

点评：请参看上文第 38 题第 4 小题：9 月 26 日，百度一二面试题。

三、系统设计题（30 分）

RSA SecurID 安全系统

应用场景：这是一种用户登录验证手段，例如银行登录系统，这个设备显示 6 位数字，每 60 秒变一次，再经过服务器认证，通过则允许登录。问 How to design this system?

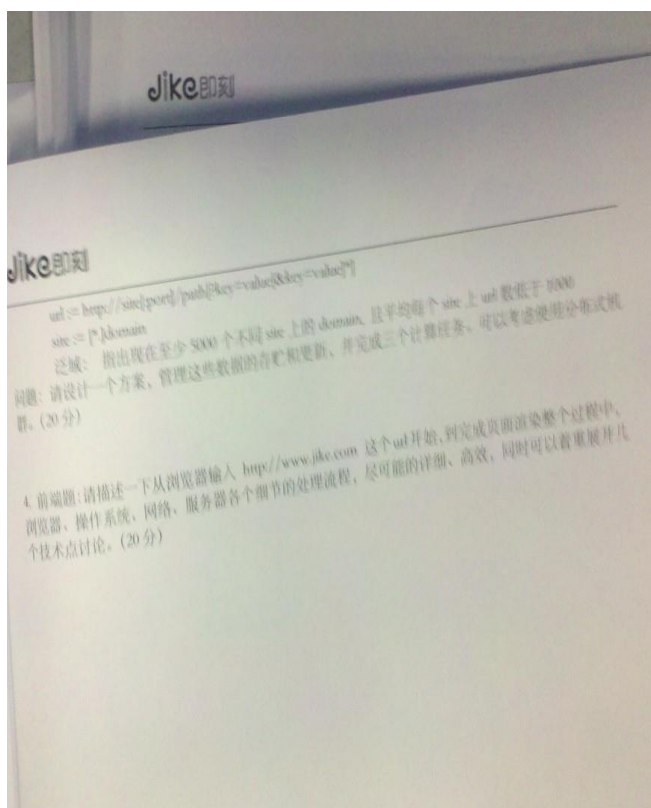
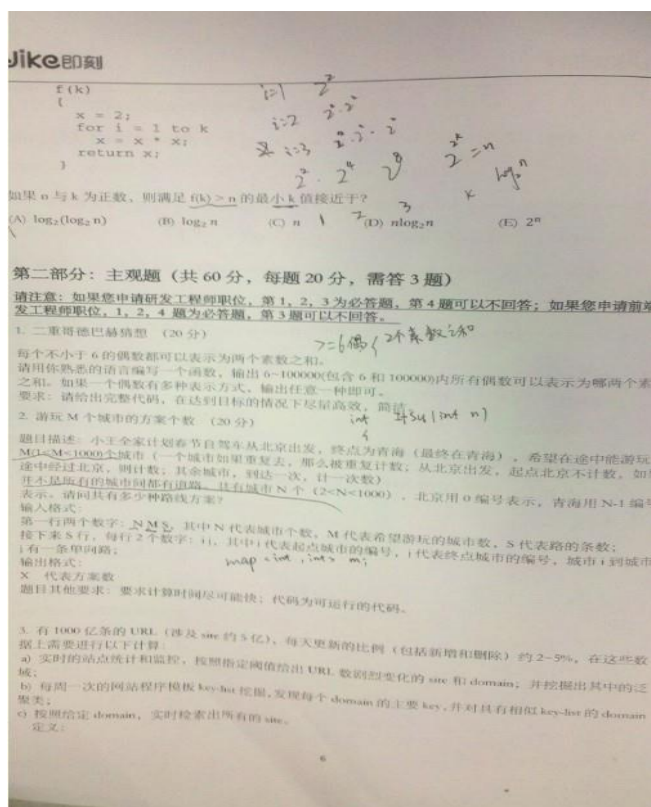
- 1) 系统设计思路? 服务器端为何能有效认证动态密码的正确性?
- 2) 如果是千万量级永固, 给出系统设计图示或说明, 要求子功能模块划分清晰, 给出关键的数据结构或数据库表结构。  
考虑用户量级的影响和扩展性, 用户密码的随机性等, 如果设计系统以支持这几个因素.
- 3) 系统算法升级时, 服务器端和设备端可能都要有所修改, 如何设计系统, 能够使升级过程(包括可能的设备替换或重设)尽量平滑?

70. 10月13日, 百度移动开发笔试题

- 一、
  - 1、什么是 RISC;
  - 2、通过后序、中序求前序
  - 3、重写与重载的区别
- 二、
  - 1、反转链表
  - 2、判断两个数组中是否有相同的数字
  - 3、1000 瓶水中找出有毒的那瓶, 毒性一周后发作, 一周内最少需要多少只老鼠
- 三、系统设计 email 客户端, 支持多账户和 pop3 等协议
  - 1、请写出可能的至少 5 个用例;
  - 2、使用 sqlite 存储帐户、已收信息、已发信息、附件、草稿, 请设计合理的表结构
  - 3、pop3 等协议等接口已完成, 请给出 email 客户端的模块设计图。



71. 10月13日，人搜2013校招北京站部分笔试题（读者回忆+照片）：



## 1，二重歌德巴赫猜想

所有大于等于6的偶数都可以表示成两个（奇）素数之和。



给定 1-10000，找到可以用两个素数之和表示每一个偶数的两个素数，然后输出这两个素数，如果有多对，则只需要输出其中之一对即可。

要求：复杂度较低，代码可运行。

## 2，城市遍历

某人家住北京，想去青海玩，可能会经过许多城市，

现已知地图上的城市连接，求经过 M 个城市到达青海的路线种类。

城市可以多次到达的，比如去了天津又回到北京，再去天津，即为 3 次。北京出发不算 1 次。

输入：

N    M    S

N 为城市总数，北京为 0，青海为 N-1；

M 为经过的城市数目；

S 为之后有 S 行

i    j

表示第 i 个城市可以去第 j 个城市，是有方向的。

输出：

N

表示路径种类。

## 3，分布式系统设计

有 1000 亿个 URL，其中大约有 5 亿个 site。每天的更新大约 2%-5%。设计一个系统来解决存储和计算下面三个问题。可用分布式系统。

URL: http://site[port]\*(key==?;key==?)

site:[\*].domain

URL:

http://www.baidu.com/baidu?word=%E5%AE%A3%E8%AE%B2%E4%BC%9A&ie=utf-8

site: : www.baidu.com

domain: : baidu.com

key=baidu?word

a>检测每个域名下的 site 数目，以及每个 site 下的 URL 数目，输出 site 变化超过一定阈值的域名以及 URL 数目变化剧烈的 site。找出泛域。

泛域：该域下的 site 数目超过 500 个，且每个 site 下的 URL 数目超过 100 个。

b>提取 URL 中 key 的特征，对 site 进行聚类；

(每个 site 下面有多个 URL，这些 URL 中有许多 key，可以获取这些 key 作为 site 的特征，对 site 进行聚类，不过这应该是多机器联合的)

c>对于给定的 domain，输出该 domain 下的所有 site。

72. 10月13日，创新工场笔试：

第一个，快排最坏情况下是  $O(n^2)$ ，问如何优化？

第二个，怎么样求一个数的根号

点评：你是不是会想到一系列有关数学的东西，什么泰勒级数啊，什么牛顿法啊，具体编程可以如下代码所示：

```
1. static void Main(string[] args)
2. {
3.     double k = 5;
4.     double n = 2, m = k;
5.
6.     while (n != m)
7.     {
8.         m = k / n;
9.         n = (m + n) / 2;
10.    }
11. }
```

链接：<http://www.51nod.com/question/index.html#!questionId=660>。

第三个，4个数字，用四则元素求结果能否为24。写出这个判断的函数。

73. 10月14日，思科网讯旗下公司笔试题：

1、海量数据中，寻找最小的k个数。

请分情况，给出时间复杂度最优，或空间复杂度最优的方案，或时间复杂度/空间复杂度综合考虑的可行方案。

点评：参见：第三章、寻找最小的k个数。

2、有两座桥，其中一座可能是坏的，两个守桥人分别守在这两座桥的入口。他们一个总是会说实话，一个总是说谎话。

你现在需要找出哪一座桥可以通过。

1)、请问最少需要问守桥人几个问题，可以找出可以通过的桥？如何问？

2)、请编程解决。

74. 10月14日，腾讯杭州站笔试题：

1、http服务器会在用户访问某一个文件的时候，记录下该文件被访问的日志，网站管理员都会去统计每天每文件被访问的次数。写一个小程序，来遍历整个日志文件，计算出每个文件被访问的访问次数

1) 请问这个管理员设计这个算法

2) 该网站管理员后来加入腾讯从事运维工作，在腾讯，单台http服务器不够用的，同样的内容，会分布在全国各地上百台服务器上。每台服务器上的日志数量，

都是之前的 10 倍之多，每天服务器的性能更好，之前他用的是单核 cpu，现在用的是 8 核的，管理员发现在这种的海量的分布式服务器，基本没法使用了，请重新设计一个算法。

2、腾讯的 qq 游戏当中，最多人玩的游戏就是斗地主了，每一句游戏开始时，服务器端都要洗牌，以保证发牌的时每个人拿的牌都是随机的，假设用 1-54 来表示 54 张不同的牌，请你写一个洗牌算法，保证 54 张牌能随机打散！

选择题：

1)、下列 RAID 技术无法提高可靠性的是：

A: RAID0 B: RAID1 C: RAID10 D: RAID5

2)、长度为 1 的线段，随机在其上选择两点，将线段分为三段，问这 3 个字段能组成一个三角形的概率是：

1/2,1/3,1/4,1/8

3)、下面那种标记的包不会在三次握手的过程中出现 ( )

A: SYN B: PSH C: ACK D: RST

75. 10 月 14 日，搜狗 2013 校招笔试题：

1、有  $n \times n$  个格子，每个格子里有正数或者 0，从最左上角往最右下角走，只能向下和向右，一共走两次（即从左下角走到右下角走两趟），把所有经过的格子的数加起来，求最大值 SUM，且两次如果经过同一个格子，则最后总和 SUM 中该格子的计数只加一次。

点评：@西芹\_new，一共搜  $(2n-2)$  步，每一步有四种走法，考虑不相交等条件可以剪去很多枝，代码如下「<http://blog.csdn.net/huangxy10/article/details/8071242>」：

```
1. 西芹_new<huangxy10@qq.com> 0:55:40
2. // 10_15.cpp : 定义控制台应用程序的入口点。
3. //
4.
5. #include "stdafx.h"
6. #include <iostream>
7. using namespace std;
8.
9. #define N 5
10. int map[5][5]={
11.     {2,0,8,0,2},
12.     {0,0,0,0,0},
13.     {0,3,2,0,0},
14.     {0,0,0,0,0},
15.     {2,0,8,0,2}};
16. int sumMax=0;
17. int p1x=0;
```

```

18. int p1y=0;
19. int p2x=0;
20. int p2y=0;
21. int curMax=0;
22.
23. void dfs( int index){
24.     if( index == 2*N-2){
25.         if( curMax>sumMax)
26.             sumMax = curMax;
27.         return;
28.     }
29.
30.     if( !(p1x==0 && p1y==0) && !(p2x==N-1 && p2y==N-1))
31.     {
32.         if( p1x>= p2x && p1y >= p2y )
33.             return;
34.     }
35.
36.     //right right
37.     if( p1x+1<N && p2x+1<N ){
38.         p1x++;p2x++;
39.         int sum = map[p1x][p1y]+map[p2x][p2y];
40.         curMax += sum;
41.         dfs(index+1);
42.         curMax -= sum;
43.         p1x--;p2x--;
44.     }
45.
46.     //down down
47.     if( p1y+1<N && p2y+1<N ){
48.         p1y++;p2y++;
49.         int sum = map[p1x][p1y]+map[p2x][p2y];
50.         curMax += sum;
51.         dfs(index+1);
52.         curMax -= sum;
53.         p1y--;p2y--;
54.     }
55.
56.     //rd
57.     if( p1x+1<N && p2y+1<N ) {
58.         p1x++;p2y++;
59.         int sum = map[p1x][p1y]+map[p2x][p2y];
60.         curMax += sum;
61.         dfs(index+1);

```

```

62.         curMax -= sum;
63.         p1x--;p2y--;
64.     }
65.
66.     //dr
67.     if( p1y+1<N && p2x+1<N ) {
68.         p1y++;p2x++;
69.         int sum = map[p1x][p1y]+map[p2x][p2y];
70.         curMax += sum;
71.         dfs(index+1);
72.         curMax -= sum;
73.         p1y--;p2x--;
74.     }
75. }
76.
77. int _tmain(int argc, _TCHAR* argv[])
78. {
79.     curMax = map[0][0];
80.     dfs(0);
81.     cout <<sumMax-map[N-1][N-1]<<endl;
82.     return 0;
83. }

```

@绿色夹克衫：跟这个问题：

<http://www.51nod.com/question/index.html#!questionId=487>，是同一个问题。

1、用动态规划可以求解，大概思路就是同时 DP 2 次所走的状态。先来分析一下这个问题，为了方便讨论，先对矩阵做一个编号，且以 5\*5 的矩阵为例（给这个矩阵起个名字叫 M1）：

M1

0 1 2 3 4

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

从左上(0)走到右下(8)共需要走 8 步 ( $2*5-2$ )。为了方便讨论,我们设所走的步数为  $s$ 。因为限定了只能向右和向下走,因此无论如何走,经过 8 步后 ( $s = 8$ )都将走到右下。而 DP 的状态也是依据所走的步数来记录的。

再来分析一下经过其他  $s$  步后所处的位置,根据上面的讨论,可以知道经过 8 步后,一定处于右下角(8),那么经过 5 步后( $s = 5$ ),肯定会处于编号为 5 的位置。3 步后肯定处于编号为 3 的位置.....。  $s = 4$  的时候,处于编号为 4 的位置,对于方格中,共有 5 (相当于  $n$ ) 个不同的位置,也是所有编号中最多的。推广来说  $n*n$  的方格,总共需要走  $2n - 2$  步,当  $s = n - 1$  时,编号为  $n$  个,也是编号最多的。

如果用  $DP[s,i,j]$ 来记录 2 次所走的状态获得的最大值,其中  $s$  表示走  $s$  步,  $i$  表示  $s$  步后第 1 次走所处的位置,  $j$  表示  $s$  步后第 2 次走所处的位置。

为了方便讨论,再对矩阵做一个编号(给这个矩阵起个名字叫 M2):

M2

0 0 0 0 0

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

M1

0 1 2 3 4

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

经过 6 步后，肯定处于 M1 中编号为 6 的位置。共有 3 个编号为 6 的，分别对应 M2 中的 2 3 4。假设第 1 次经过 6 步走到了 M2 中的 2，第 2 次经过 6 步走到了 M2 中的 4， $DP[s,i,j]$  则对应  $DP[6,2,4]$ 。由于  $s = 2n - 2, 0 \leq i \leq j \leq n$ ，所以这个 DP 共有  $O(n^3)$  个状态。

M1

0 1 2 3 4

1 2 3 4 5

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

再分析一下状态转移，以  $DP[6,2,3]$  为例(就是上面 M1 中加粗的部分)，可以到达  $DP[6,2,3]$  的状态包括  $DP[5,1,2]$ ， $DP[5,1,3]$ ， $DP[5,2,2]$ ， $DP[5,2,3]$ ，加粗表示位置  $DP[5,1,2]$   $DP[5,1,3]$   $DP[5,2,2]$   $DP[5,2,3]$ （加红表示要达到的状态  $DP[6,2,3]$ ）

0 1 2 3 4	0 1 2 3 4	0 1 2 3 4	0 1 2 3 4
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 3 4 5 6	2 3 4 5 6	2 3 4 5 6	2 3 4 5 6
3 4 5 6 7	3 4 5 6 7	3 4 5 6 7	3 4 5 6 7
4 5 6 7 8	4 5 6 7 8	4 5 6 7 8	4 5 6 7 8

因此， $DP[6,2,3] = \text{Max}(DP[5,1,2], DP[5,1,3], DP[5,2,2], DP[5,2,3]) + 6, 2$  和  $6, 3$  格子中对应的数值（式一）

2、上面（式一）所示的这个递推看起来没有涉及：“如果两次经过同一个格子，那么该数只加一次的这个条件”，讨论这个条件需要换一个例子，以  $DP[6,2,2]$  为例。

$DP[6,2,2]$  可以由  $DP[5,1,1]$ ， $DP[5,1,2]$ ， $DP[5,2,2]$  到达，但由于  $i = j$ ，也就是 2 次走到同一个格子，那么数值只能加 1 次。

所以当  $i = j$  时， $DP[6,2,2] = \text{Max}(DP[5,1,1], DP[5,1,2], DP[5,2,2]) + 6, 2$  格子中对应的数值（式二）

3、故，综合上述的（式一），（式二）最后的递推式就是



if(i != j)

$DP[s, i, j] = \text{Max}(DP[s - 1, i - 1, j - 1], DP[s - 1, i - 1, j], DP[s - 1, i, j - 1], DP[s - 1, i, j]) + W[s, i] + W[s, j]$

else

$DP[s, i, j] = \text{Max}(DP[s - 1, i - 1, j - 1], DP[s - 1, i - 1, j], DP[s - 1, i, j]) + W[s, i]$

其中  $W[s, i]$  表示经过  $s$  步后，处于  $i$  位置，位置  $i$  对应的方格中的数字。

复杂度分析：状态转移最多需要统计 4 个变量的情况，看做是  $O(1)$  的。共有  $O(n^3)$  个状态，所以总的时间复杂度是  $O(n^3)$  的。空间上可以利用滚动数组优化，由于每一步的递推只跟上 1 步的情况有关，因此可以循环利用数组，将空间复杂度降为  $O(n^2)$ 。

OK，上述这个方法可能不算最优解法，但相对比较容易想一些。希望大家能够提供更好的想法，也欢迎大家补充程序。链接：

<http://www.51nod.com/answer/index.html#!answerId=598>。

2、N 个整数（数的大小为 0-255）的序列，把它们加密为 K 个整数（数的大小为 0-255）。再将 K 个整数顺序随机打乱，使得可以从这乱序的 K 个整数中解码出原序列。设计加密解密算法，且要求  $K \leq 15 * N$ 。

如果是：

1， $N \leq 16$ ，要求  $K \leq 16 * N$ 。

2， $N \leq 16$ ，要求  $K \leq 10 * N$ 。

3， $N \leq 64$ ，要求  $K \leq 15 * N$ 。

点评：<http://www.51nod.com/question/index.html#!questionId=659>。

76. 人人网面试，只面一道题，要求 5 分钟出思路，10 分钟出代码

面试题是：

两个无序数组分别叫 A 和 B，长度分别是  $m$  和  $n$ ，求中位数，要求时间复杂度  $O(m+n)$ ，空间复杂度  $O(1)$ 。

77. 10 月 15 日，网新恒天笔试题

1. 不要使用库函数，写出 `void *memcpy(void *dst, const void *src, size_t count)`，其中 `dst` 是目标地址，`src` 是源地址。

点评：下面是 [nwpulei](#) 写的代码：

```

1. void* memcpy(void *dst, const void *src, size_t count)
2. {
3.     assert(dst != NULL);
4.     assert(src != NULL);
5.     unsigned char *pdst = (unsigned char *)dst;
6.     const unsigned char *psrc = (const unsigned char *)src;
7.
8.     assert(!(psrc<=pdst && pdst<psrc+count));
9.     assert(!(pdst<=psrc && psrc<pdst+count));
10.
11.     while(count--)
12.     {
13.         *pdst = *psrc;
14.         pdst++;
15.         psrc++;
16.     }
17.     return dst;
18. }

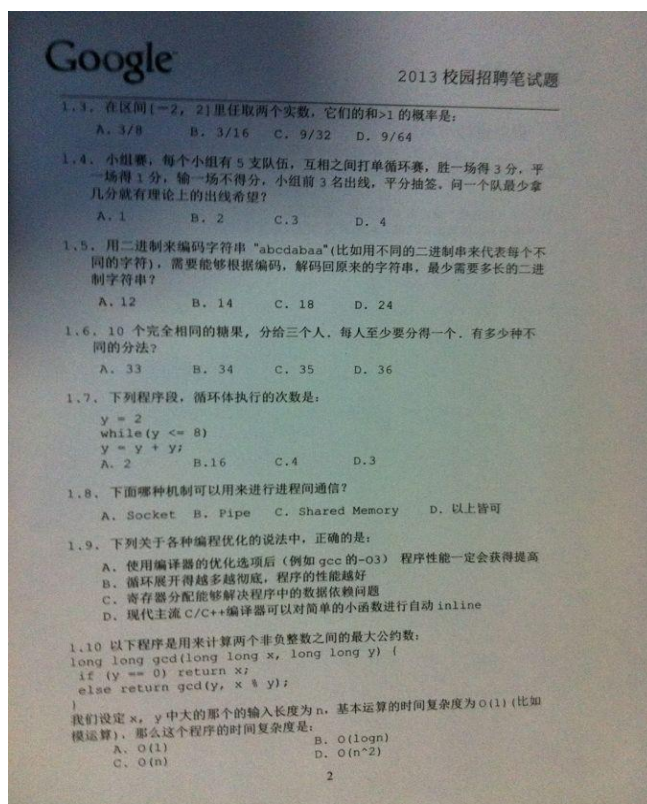
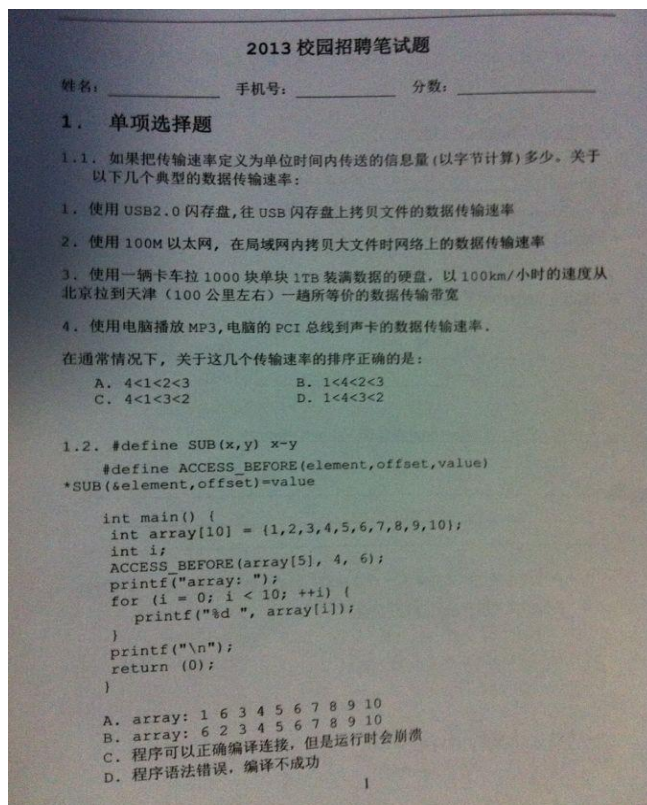
```

链接: <http://blog.csdn.net/nwpulei/article/details/8090136>。

2. 给定一个字符串, 统计一下哪个字符出现次数最大。

3. 我们不知道 Object 类型的变量里面会出现什么内容, 请写个函数把 Object 类型转换为 int 类型。

78. 10月15日, Google 2013 校招全套笔试题:



Google

2013 校园招聘笔试题

## 2. 程序设计与算法

(2.1, 2.2 为编程题, 需要写出程序实现; 2.3 为算法设计题, 只需要给出算法设计思路及关键步骤的伪代码即可。)

2.1. 写一个函数, 输出前  $N$  个素数。函数原型: `void print_prime(int N);`  
不需要考虑整数溢出问题, 也不需要使用大数处理算法。

Google

2013 校园招聘笔试题

2.2. 长度为  $N$  的数组乱序存放着  $0$  到  $N-1$ 。现在只能进行  $0$  与其他数的 swap, 请设计并实现排序。

Google

2013 校园招聘笔试题

2.3. 给定一个源串和目标串, 能够对源串进行如下操作:

- 1 在给定位置上插入一个字符
- 2 替换任意字符
- 3 删除任意字符

要求写一个程序, 返回最少的操作数, 使得对源串进行这些操作后等于目标串。  
源串和目标串的长度都小于 2000。

1. 写一个函数, 输出前  $N$  个素数, 函数原型: `void print_prime(int N);` 不需要考虑整数的溢出问题, 也不需要使用大数处理算法。

2. 长度为  $N$  的数组乱序存放着  $0$  到  $N-1$ 。现在只能进行  $0$  与其他数的 swap 操作, 请设计并实现排序, 必须通过交换实现排序。

3. 给定一个源串和目标串, 能够对源串进行如下操作:

1. 在给定位置上插入一个字符
2. 替换任意字符
3. 删除任意字符

写一个程序，返回最小操作数，使得对源串进行这些操作后等于目标串，源串和目标串的长度都小于 2000。

点评：

1、此题反复出现，如上文第 38 题第 4 小题 9 月 26 日百度一二面试题，10 月 9 日腾讯面试题第 1 小题，及上面第 69 题 10 月 13 日百度 2013 校招北京站笔试题第二大道题第 3 小题，同时，还可以看下这个链接：

<http://www.51nod.com/question/index.html#!questionId=662>。

```
1. //动态规划：
2.
3. //f[i,j]表示 s[0...i]与 t[0...j]的最小编辑距离。
4. f[i,j] = min { f[i-1,j]+1, f[i,j-1]+1, f[i-1,j-1]+(s[i]==t[j]?0:1) }
5.
6. //分别表示：添加 1 个，删除 1 个，替换 1 个（相同就不用替换）。
```

2、补充：上述问题类似于编程之美上的下述一题「以下内容摘自编程之美第 3.3 节」：许多程序会大量使用字符串。对于不同的字符串，我们希望能够有办法判断其相似程度。我们定义了一套操作方法来把两个不相同的字符串变得相同，具体的操作方法为：

1. 修改一个字符（如把“a”替换为“b”）；
2. 增加一个字符（如把“abdd”变为“aebdd”）；
3. 删除一个字符（如把“travelling”变为“traveling”）。

比如，对于“abcdefg”和“abcdef”两个字符串来说，我们认为可以通过增加/减少一个“g”的方式来达到目的。上面的两种方案，都仅需要一次操作。把这个操作所需要的次数定义为两个字符串的距离，而相似度等于“距离+1”的倒数。也就是说，“abcdefg”和“abcdef”的距离为 1，相似度为  $1 / 2 = 0.5$ 。

给定任意两个字符串，你是否能写出一个算法来计算出它们的相似度呢？



不难看出，两个字符串的距离肯定不超过它们的长度之和（我们可以通过删除操作把两个串都转化为空串）。虽然这个结论对结果没有帮助，但至少可以知道，任意两个字符串的距离都是有限的。

考虑如何才能把这个问题转化成规模较小的同样的问题。如果有两个串  $A = xabcdae$  和  $B = xfdfa$ ，它们的第一个字符是相同的，只要计算  $A[2, \dots, 7] = abcdae$  和  $B[2, \dots, 5] = fdfa$  的距离就可以了。但是如果两个串的第一个字符不相同，那么可以进行如下的操作（ $lenA$  和  $lenB$  分别是  $A$  串和  $B$  串的长度）。

1. 删除  $A$  串的第一个字符，然后计算  $A[2, \dots, lenA]$  和  $B[1, \dots, lenB]$  的距离。
2. 删除  $B$  串的第一个字符，然后计算  $A[1, \dots, lenA]$  和  $B[2, \dots, lenB]$  的距离。
3. 修改  $A$  串的第一个字符为  $B$  串的第一个字符，然后计算  $A[2, \dots, lenA]$  和  $B[2, \dots, lenB]$  的距离。
4. 修改  $B$  串的第一个字符为  $A$  串的第一个字符，然后计算  $A[2, \dots, lenA]$  和  $B[2, \dots, lenB]$  的距离。
5. 增加  $B$  串的第一个字符到  $A$  串的第一个字符之前，然后计算  $A[1, \dots, lenA]$  和  $B[2, \dots, lenB]$  的距离。
6. 增加  $A$  串的第一个字符到  $B$  串的第一个字符之前，然后计算  $A[2, \dots, lenA]$  和  $B[1, \dots, lenB]$  的距离。

在这个题目中，我们并不在乎两个字符串变得相等之后的字符串是怎样的。所以，可以将上面 6 个操作合并为：

1. 一步操作之后，再将  $A[2, \dots, lenA]$  和  $B[1, \dots, lenB]$  变成相同字符串；
2. 一步操作之后，再将  $A[1, \dots, lenA]$  和  $B[2, \dots, lenB]$  变成相同字符串；
3. 一步操作之后，再将  $A[2, \dots, lenA]$  和  $B[2, \dots, lenB]$  变成相同字符串。

这样，很快就可以完成一个递归程序，如下所示：

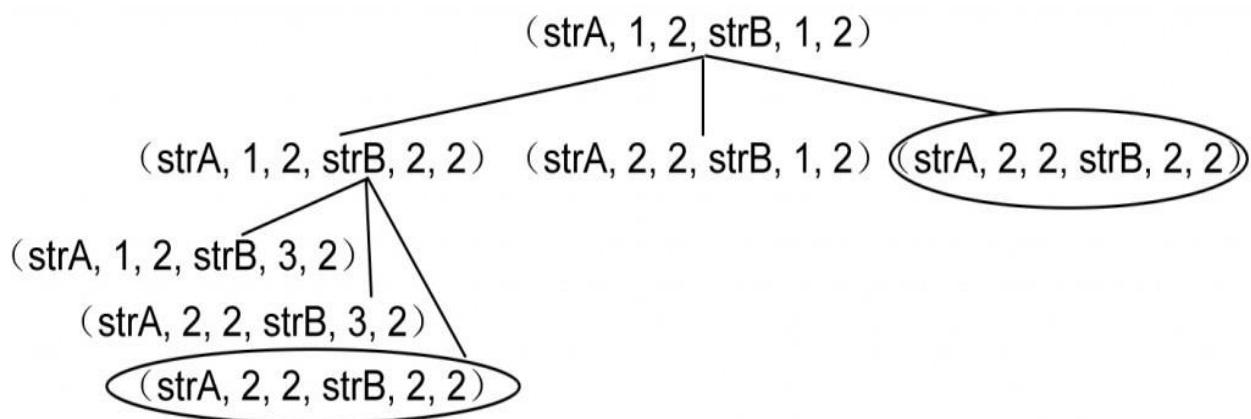
```
1. Int CalculateStringDistance(string strA, int pABegin, int pAEnd,
2.    string strB, int pBBegin, int pBEnd)
3. {
4.     if(pABegin > pAEnd)
5.     {
6.         if(pBBegin > pBEnd)
7.             return 0;
8.         else
9.
10.            return pBEnd - pBBegin + 1;
11.     }
12.
13.     if(pBBegin > pBEnd)
14.     {
15.         if(pABegin > pAEnd)
16.             return 0;
```

```

17.         else
18.             return pAEnd - pABegin + 1;
19.     }
20.
21.     if(strA[pABegin] == strB[pBBegin])
22.     {
23.         return CalculateStringDistance(strA, pABegin + 1, pAEnd,
24.             strB, pBBegin + 1, pBEnd);
25.     }
26.     else
27.     {
28.         int t1 = CalculateStringDistance(strA, pABegin, pAEnd, strB,
29.             pBBegin + 1, pBEnd);
30.         int t2 = CalculateStringDistance(strA, pABegin + 1, pAEnd,
31.             strB, pBBegin, pBEnd);
32.         int t3 = CalculateStringDistance(strA, pABegin + 1, pAEnd,
33.             strB, pBBegin + 1, pBEnd);
34.         return minValue(t1, t2, t3) + 1;
35.     }
36. }

```

上面的递归程序，有什么地方需要改进呢？在递归的过程中，有些数据被重复计算了。比如，如果开始我们调用 `CalculateStringDistance(strA, 1, 2, strB, 1, 2)`，下图是部分展开的递归调用。



可以看到，圈中的两个子问题被重复计算了。为了避免这种不必要的重复计算，可以把子问题计算后的解存储起来。如何修改递归程序呢？还是 DP！请看此链接：

<http://www.cnblogs.com/yujunyong/articles/2004724.html>。

3、此外，关于这个“编辑距离”问题的应用：搜索引擎关键字查询中拼写错误的提示，可以看下这篇文章：[http://www.ruanyifeng.com/blog/2012/10/spelling\\_corrector.html](http://www.ruanyifeng.com/blog/2012/10/spelling_corrector.html)。「关于什么

是“编辑距离”：一个快速、高效的 **Levenshtein** 算法实现，这个是计算两个字符串的算法，**Levenshtein** 距离又称为“编辑距离”，是指两个字符串之间，由一个转换成另一个所需的最少编辑操作次数。当然，次数越小越相似。这里有一个 **BT** 树的数据结构，挺有意思的：<http://blog.notdot.net/2007/4/Damn-Cool-Algorithms-Part-1-BK-Trees>」最后，**Lucene** 中也有这个算法的实现（我想，一般的搜索引擎一般都应该会有此项拼写错误检查功能的实现）：<http://www.bjwilly.com/archives/395.html>。

**4、扩展：**面试官还可以继续问下去：那么，请问，如何设计一个比较两篇文章相似性的算法？（这个问题的讨论可以看看这里：<http://t.cn/zl82CAH>）

**BTW**，群友 **braveheart89** 也整理了这套笔试题：

<http://blog.csdn.net/braveheart89/article/details/8074657>。

**79.** 10 月 16 日，**UC** 的笔试题目：

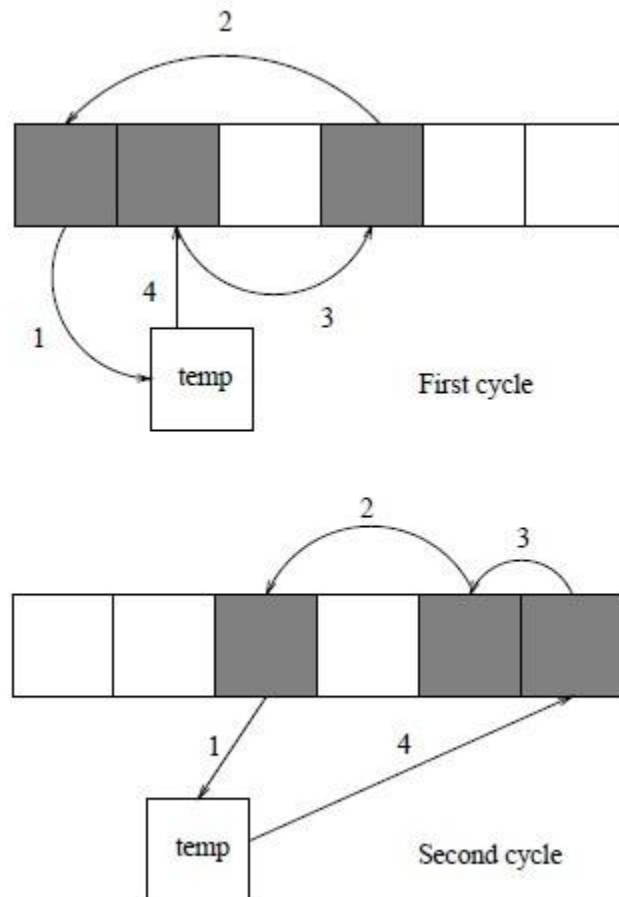
1、有个长度为  $2n$  的数组  $\{a_1, a_2, a_3, \dots, a_n, b_1, b_2, b_3, \dots, b_n\}$ ，希望排序后  $\{a_1, b_1, a_2, b_2, \dots, a_n, b_n\}$ ，要求时间复杂度  $O(n)$ ，空间复杂度  $O(1)$ 。

点评：@绿色夹克衫：完美洗牌问题「关于洗牌算法：

<http://blog.csdn.net/gogdizzy/article/details/4917488>」，这个问题的关键在于如何解决置换群中的环。方法是微软员工那篇论文中写的：

<http://user.qzone.qq.com/414353346/blog/1243343118#!app=2&via=QZ.HashRefresh&pos=1243343118>，大概意思是，用 3 的幂来弄：





@方程：

```

1. int index = arr.length / 2;
2. int temp = arr[index];
3. while(index != 1){
4.     int tempIndex = (index + (index % 2) * (arr.length - 1)) / 2;

5.     arr[index] = arr[tempIndex];
6.     index = tempIndex;
7. }
8. arr[1] = temp;

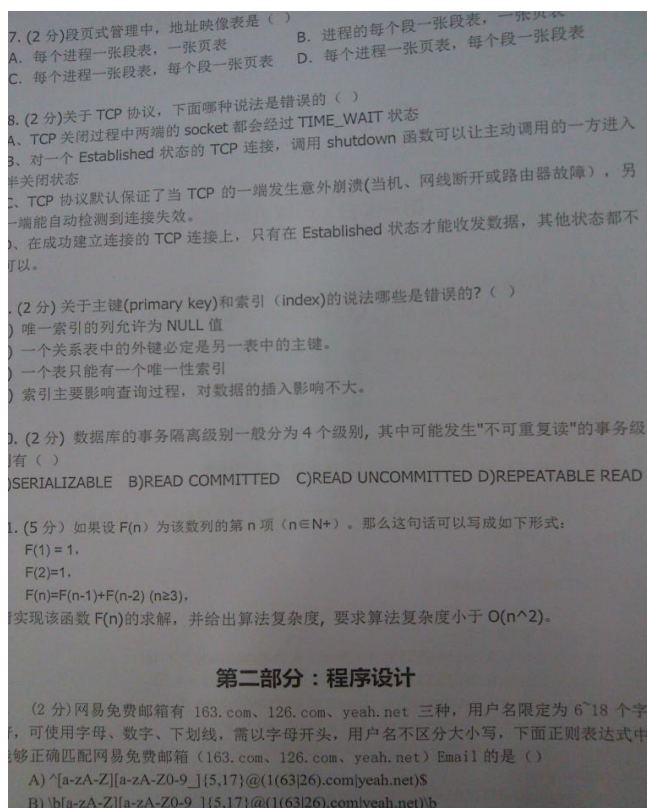
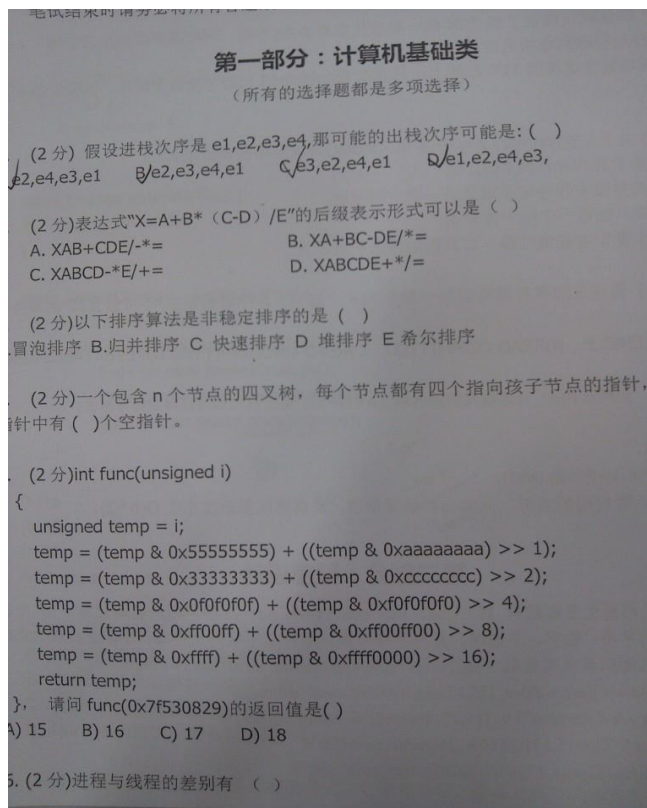
```

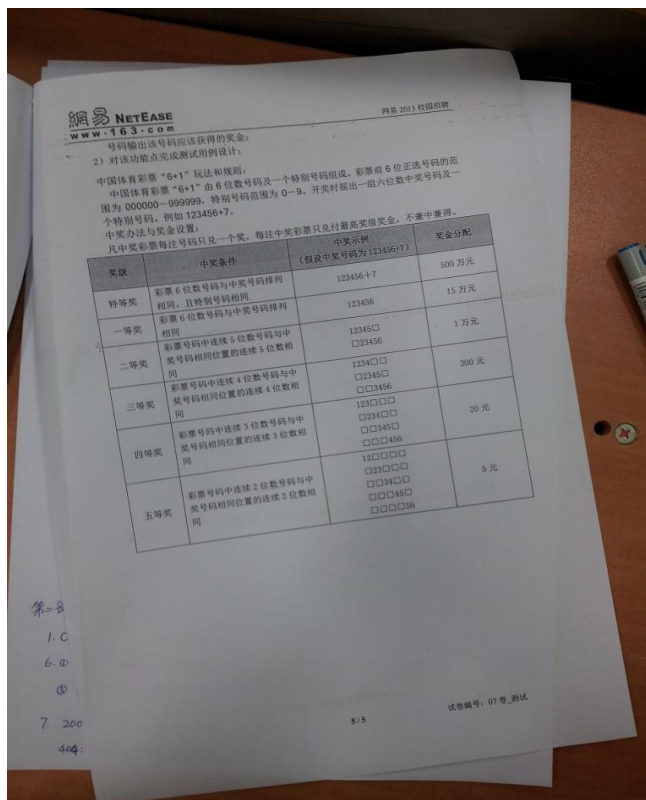
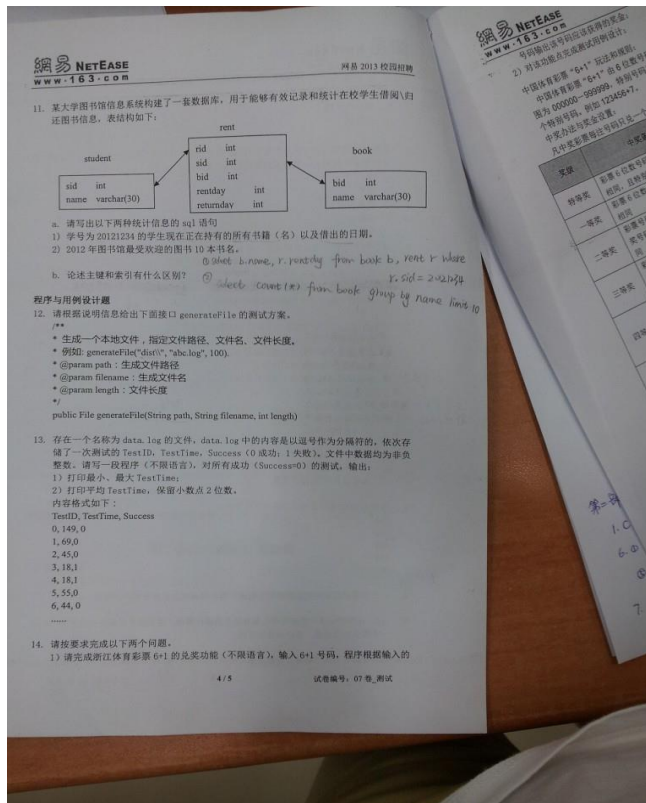
链接：1，<http://www.51nod.com/question/index.html#!questionId=278>；2、这里也有一参考答案：<http://blog.csdn.net/yuan8080/article/details/5705567>。

80. 10月17日，创新工场电话面试：

- 1，如何删除一个搜索二叉树的结点；
- 2，如何找到一个数组中的两个数，他们的和为0；
- 3，如何判断两条二维平面上的线段是否相交。

## 81. 网易 2013 校招笔试题:





82. 10 月 19 日，百度研发三面题：

百度地图里的路线查询：给定两个站点，如果没有直达的路线，如何找到换乘次数

最少的路线？

点评：蚂蚁算法？还是广搜，或 A\* 算法？

83. 10 月 20 日，baidu 广州站笔试算法题：

1. 有一箱苹果，3 个一包还剩 2 个，5 个一包还剩 3 个，7 个一包还剩 2 个，求 N 个满足以上条件的苹果个数。
2. 用递归算法写一个函数，求字符串最长连续字符的长度，比如 aaaabbcc 的长度为 4，aabb 的长度为 2，ab 的长度为 1。
3. 假设一个大小为 100 亿个数据的数组，该数组是从小到大排好序的，现在该数组分成若干段，每个段的数据长度小于 20「也就是说：题目并没有说每段数据的 size 相同，只是说每个段的 size < 20 而已」，然后将每段的数据进行乱序（即：段内数据乱序），形成一个新数组。请写一个算法，将所有数据从小到大进行排序，并说明时间复杂度。

点评：

思路一、如 @四万万网友 所说：维护一个 20 个元素大小的小根堆，然后排序，每次 pop 取出小根堆上最小的一个元素 ( $\log 20$ )，然后继续遍历原始数组后续的 (N-20) 个元素，总共 pop (N-20) 次 20 个元素小根堆的  $\log 20$  的调整操作。

思路二 @飘零虾、如果原数组是 a[]，那么  $a[i+20] \geq a[i]$  恒成立（因为每段乱序区间都是小于 20 的，那么向后取 20，必然是更大的区间的元素）。

第一个数组：取第 0、20、40、60、80...

第二个数组：取第 1、21、41、61、81...

...

第 20 个数组：取第 19、39、59、79... （上述每个数组 100 亿/20 个元素）

共计 20 个数组，每个数组 100 亿/20 个元素「注：这 5 亿个元素已经有序，不需要再排序」，且这 20 个数组都是有序的，然后对这 20 个数组进行归并，每次归并 20 个元素。时间复杂度跟上述思路一一样，也是  $N \cdot \log K$  (N=100 亿，K=20)。

此外，读者 @木叶漂舟 直接按每组 20 个排序，将排好的 20 个与前 20 个调整拼接，调整两端接头处的元素，写了个简单地 demo：<http://t.cn/zLELAzs>。不过，复杂度有点高，目前来说中规中矩的思路还是如上文中 @四万万网友 所说思路一「@张玮-marihees 按照思路一：<http://weibo.com/1580904460/z1v5jxJ9P>，写了一份代码：

<http://codepad.org/T5jIUFP6>，欢迎查看」。

84. 10 月 21 日，完美笔试算法题「同时，祝自己生日快乐！」：

1. 请设计一个算法，当给出在 2D 平面中某个三角形 ABC 的顶点坐标时能输出位于该三角形内的一个随机点(需要满足三角形内均匀随机)，无需写出实现，只要能清楚地描述算法即可。

2. 请自己用双向链表实现一个队列，队列里节点内存的值为 `int`，要求实现入队，出队和查找指定节点的三个功能。

3. 实现一个无符号任意大整数的类，实现两个无符号超大整数的乘法。

85. 10月22日，CSR 掌微电子笔试题：

5. 给定两个字符串 `s1` 和 `s2`，要求判定 `s2` 是否能够通过 `s1` 做循环移位 (`rotate`) 得到字符串包含。例如，`S1=AABCD` 和 `s2=CDAA`，返回 `true`；给定 `s1=ABCD` 和 `s2=ACBD`，返回 `false`。用伪代码或流程图叙述解法。

点评：老题，类似：[http://blog.csdn.net/v\\_JULY\\_v/article/details/6322882](http://blog.csdn.net/v_JULY_v/article/details/6322882)。其余题目见：[http://blog.sina.com.cn/s/blog\\_3eb9f728010161lt.html](http://blog.sina.com.cn/s/blog_3eb9f728010161lt.html)。

86. 10月23日，去哪儿网笔试：

1. 将 `IPV4` 转换成整数

2. 定义一个栈的数据结构，实现 `min` 函数，要求 `push`, `pop`, `min` 时间复杂度是  $O(1)$ ；

点评：这是 2010 年整理的微软 100 题的第 2 题，

[http://blog.csdn.net/v\\_JULY\\_v/article/details/6057286](http://blog.csdn.net/v_JULY_v/article/details/6057286)，答案见此文第 2 题：

[http://blog.csdn.net/v\\_JULY\\_v/article/details/6126406](http://blog.csdn.net/v_JULY_v/article/details/6126406)。

3. 数组 `a[n]` 里存有 1 到 `n` 的所有数，除了一个数 `removed`，找出这个 `missing` 的数。

4. 找出字符串中的最长子串，要求子串不含重复字符，并分析时间复杂度。

87. 10月28日，微软三面题「顺祝，老妈明天生日快乐！」：

找一个点集中与给定点距离最近的点，同时，给定的二维点集都是固定的，查询可能有很多次，时间复杂度  $O(n)$  无法接受，请设计数据结构和相应的算法。

类似于 @陈利人：附近地点搜索，就是搜索用户附近有哪些地点。随着 `GPS` 和带有 `GPS` 功能的移动设备的普及，附近地点搜索也变得炙手可热。在庞大的地理数据库中搜索地点，索引是很重要的。但是，我们的需求是搜索附近地点，例如，坐标 (39.91, 116.37) 附近 500 米内有什么餐馆，那么让你来设计，该怎么做？

点评：我看到这道题的时候，除了想到用 `R 树`「从 `B 树`、`B+树`、`B*树` 谈到 `R 树`」解决这个问题之外，还想起了之前一直要写的 `KD 树` 仍未写，但估计快要写了，请读者朋友们耐心等待些时日吧。

88. 11月10日，百度笔试题：

1、20 个排序好的数组，每个数组 500 个数，按照降序排序好的，让找出 500 个最大的数。

2、一在线推送服务，同时为 10 万个用户提供服务，对于每个用户服务从 10 万首歌的曲库中为他们随机选择一首，同一用户不能推送重复的，设计方案，内存尽可能小，写出数据结构与算法。

89.

90. 从今天开始，在继续整理笔试面试题的同时，将整理上面已经收录的一系列笔试面试题的答案，欢迎诸君与我共同讨论.思考.做之「参与的方式为：你除了可以直接评论在本文之下,你也可以通过邮件：zhoulei0907@yahoo.cn 或私信：<http://weibo.com/julyweibo> 给我，或自己写一篇博文把链接发给我收录,无任何语言限制」2012.10.19...