



[Google](#) 笔试是没有门槛的。这样说是因为 Google 根本没有限制笔试的人数，开了 N 个教室，让 N 多人参加……不过笔试本身却有门槛，看了题目就知道。

本来想上午写写的，但是，嗯，出于攒人品的目的，还是等到现在才写——现在，面试通知已经发过，很显然我又被无视了……OK，那也不错，我也没怎么准备这些东西呢，倒不是说我不重视，而是事情太多……唔，多少算是一种经验了。

回来说说昨天的笔试。题目的量并不大，除了几个单选题，剩下就是三个编程或算法题。单选就不说了，考得比较基础，涉及 C 语言常识、数据结构、文法、操作系统，主要说说大题。

大题虽然题型不一，但都有一个重要特点：考递归。精确点说，我每一题都用到了递归。

第一个的题目（嗯，记的不是很完整）：

在一棵（排序？）二叉树中搜索指定值，数据结构定义为（唉唉，数据结构的具体名字都不记得了，my god）：

```
struct Node
{
    Node * lnext;
    Node * rnext;
    int value;
};
```

函数定义为（情况同上，啥都记不清了）：

```
Node * search(Node * root, int value)
{
}
```

实现这个 **search** 函数。

用递归，经典的树的遍历，**pass** 先。

第二个的题目：

计算 Tribonaci 队列（嗯，九成九记错了那个单词……），规则是 $T(n) = T(n - 1) + T(n - 2) + T(n - 3)$ ，其中 $T(0) = T(1) = 1$ ， $T(2) = 2$ 。

函数定义：

```
int Tribonaci(int n) {
```

备注，不考虑证整数溢出，尽可能优化算法。

这一题我一看就知道要考什么，很显然的递归定义，但也是很显然的，这里所谓的优化是指**不要重复计算**。

简单的说，在计算 $T(n)$ 的时候要用到 $T(n - 1)$ 、 $T(n - 2)$ 和 $T(n - 3)$ 的结果，在计算 $T(n - 1)$ 的时候也要用到 $T(n - 2)$ 和 $T(n - 3)$ 的结果，所以在各项计算的

时候必须把以前计算的结果记录下来，去掉重复计算。这里用到的一点小技巧就是要新写一个函数用来做这种事情，嗯，看看我写的代码吧！

```
/**  
 * Get the value of T(n - 1), and retrieve the result of  
 * T(n - 2) and T(n - 3).  
 * @param[in] n The n in T(n).  
 * @param[out] mid Value of T(n - 2).  
 * @param[out] right Value of T(n - 3).  
 * @return Value of T(n - 1).  
 */  
int find_trib(int n, int & mid, int & right)  
{  
    if (3 == n)  
    {  
        mid = 1;  
        right = 1;  
        return 2;  
    }  
    else  
    {  
        int temp;  
        mid = find_trib(n - 1, right, temp);  
        return mid + right + temp;  
    }  
}  
/**  
 * Find value of T(n).  
 * @param[in] The n in T(n).  
 * @return Value of T(n).  
 * @note T(n) = T(n - 1) + T(n - 2) + T(n - 3) (n > 2)  
 *       T(0) = T(1) = 1, T(2) = 2.  
 */  
int tribonacci(int n)  
{  
    if (n < 0)  
    {
```

```
// Undefined feature.  
return 0;  
}  
  
if (0 == n || 1 == n)  
{  
    return 1;  
}  
  
if (2 == n)  
{  
    return 2;  
}  
  
int mid, right;  
int left = find_trib(n, mid, right);  
return left + mid + right;  
}
```

啊啊，对了，答卷的时候我可没心情写注释.....刚才到 vc.Net 2003 上
测试了一下，貌似没有啥问题。唉，看来我多少还是懂一点算法的.....

第三个的题目：

在一个无向图中，寻找是否有一条距离为 K 的路径，描述算法即可，不
用实现，分析算法的时间和空间复杂度，尽量优化算法。

OK，这个就是传说中的软肋了.....我也就不把自己的答案写出来
了（丢人啊），虽然后来仔细想想，我那个挫挫的方法也能够用.....只是效
率.....

That's all.

粗体文字

这都已经是昨天的事啦。之所以起这个标题是想有朝一日本博的文章也会被
搜索引擎搜到，然后访问量就是指数级增长，有没有可能啊。

话说某歌和某度居然在某一天的同一个时间搞宣讲+笔试，只不过一个在就业中心，一个在科学馆，在我 XJTU 的广袤土地上东西对峙，真是让人不记住鱼和熊掌的故事都难。Google 的笔试时间一个月前就确定了，[baidu](#) 一个周之前才得到消息，所以俺有理由认为，这是百度要问鼎中原的意思啦。够豪迈呀，就不怕人都去了 google 冷场么？看来百度还是很自信的，赞一个，况且百度的中文搜索做得不比 google 差。俺坚决支持民族自己的搜索引擎，虽然事实上俺是去了 google 笔试。此事不怪俺，想想科学馆那昏暗的灯光吧，俺觉得，非常及其适合你在台下看着你偶像的脸搞个人崇拜……

今天听说昨晚[百度](#)非常人性化，每人一瓶矿泉水，一块巧克力蛋糕，后来因为天热还每人发了纸巾擦汗，这下俺亏大了……嘿嘿。

俺本来发文的目的是说下笔试题，想想还是不说了，想知道的可以私下跟俺讨论，题目不难，全做对也不容易，不过错个两三道基本也就 kaka 了。考察得很全面，算法+数据结构+操作系统+编译原理+网络+离散数学，还居然考了个中断。

笔试之前的宣讲会，略有收获。获知 Google 全球共有员工 12000 左右，其中总部 8000 左右，而 google 中国，北京 195，上海 45，台北 35，而在一年前这一数字分别是北京 100，上海 20（这个没记准确），台北 10。我得到的唯一结论：google 中国还差的远啊，不知道开复能把它做成什么样子，应该不会撤摊子吧。

这是第二次笔试，作个记录，以备日后参考，题目另行记录。

1、 两个二进制数的异或结果

2、 递归函数最终会结束，那么这个函数一定（不定项选择）：

1. 使用了局部变量
2. 有一个分支不调用自身
3. 使用了全局变量或者使用了一个或多个参数

3、以下函数的结果？

```
int cal(int x)
{
    if(x==0)
        return 0;
    else
        return x+cal(x-1);
}
```

4、以下程序的结果？

```
void foo(int*a, int* b)
{
    *a = *a+*b;
    *b = *a-*b;
    *a = *a-*b;
}
void main()
{
    int a=1, b=2, c=3;
    foo(&a,&b);
    foo(&b,&c);
    foo(&c,&a);
    printf("%d, %d, %d", a,b,c);
}
```

5、下面哪项不是链表优于数组的特点？

1. 方便删除
2. 方便插入
3. 长度可变
4. 存储空间小

6、 $T(n) = 25T(n/5)+n^2$ 的时间复杂度？

7、 n 个顶点， m 条边的全连通图，至少去掉几条边才能构成一棵树？

8、正则表达式 $(01|10|1001|0110)^*$ 与下列哪个表达式一样？

- 1. $(0|1)^*$
- 2. $(01|01)^*$
- 3. $(01|10)^*$
- 4. $(11|01)^*$
- 5. $(01|1)^*$

9、如何减少换页错误？

1. 进程倾向于占用 CPU
2. 访问局部性（locality of reference）满足进程要求
3. 进程倾向于占用 I/O
4. 使用基于最短剩余时间（shortest remaining time）的调度机制
5. 减少页大小

10、实现两个 $N \times N$ 矩阵的乘法，矩阵由一维数组表示

11、找到单向链表中间那个元素，如果有两个则取前面一个

12、长度为 n 的整数数组，找出其中任意 $(n-1)$ 个乘积最大的那一组，只能用乘法，不可以用除法。要求对算法的时间复杂度和空间复杂度作出分析，不要求写程序。

早晨看 [SINA](#) 新闻，看到 [Google](#) 品牌价值已经达到 664.34 亿美元，跃居世界第一位。回忆昨晚陪朋友参加 [google](#) 在北大的招聘会，想和朋友们分享一些特别的感受。总体感觉这是一个无限富有，充满惊喜的公司。</div>

05 年 9 月 [google](#) 开始在北京设立公司，目前已经发展到 100 名员工。每个工程师将新配 2 台 30inch 的液晶显示器。经常到美国，澳洲，韩国，日本，印度等国家 TRAVEL，ENJOY great food and drink(喜欢吃喝玩乐)，在中国有两名外籍人士，统统讲流利的普通话。其中美国人 eric 带领的 PSO(商

务合作工程部) 部门, 9 个人, 穿着京剧戏服上班, 他扮演孙悟空, 开玩笑说穿这些工作服上班还是要花些时间的。

主要笔试考题如下, 其他题目是基础题, 就不贴出了:

1、假设在 n 进制下, 下面的等式成立, n 值是 ()

$$567 * 456 = 150216$$

- a、 9 b、 10 c、 12 d、 18

2、文法 $G: S \rightarrow uvSvu \mid w$ 所识别的语言是: ()

- a、 uvw^*vu b、 $(uvvvu)^*$ c、 $uv(uv)^*wvu(vu)^*$ d、 $(uv)^*w(vu)^*$

3、如下程序段输出是: ()

```
char str[][10]={"Hello","Google"};
```

```
char *p=str[0];
```

```
count<<strlen(p+10);
```

- a、 0 b、 5 c、 6 d、 10

4、 $cnt=0$

```
while(x!=1){  
    cnt=cnt+1;  
    if(x&1==0)  
        x=x/2;  
    else  
        x=3*x+1;  
}  
count<<cnt<<endl;
```

当 $n=11$ 时, 输出: ()

- a、 12 b、 13 c、 14 d、 15

5、写一段程序判断一个有向图 G 中节点 w 是否从节点 v 可达。 (如果 G 中存在一条从 v 至 w 的路径就说节点 w 是从 v 可达的)。以下算法是用 C++ 写成的, 在 `bool Reachable` 函数中, 你可以写出自己的算法。

```
class Graph{
public:
int NumberOfNodes();//返回节点的总数

bool HasEdge(int u,int v);//u,v 是节点个数，从零开始依次递增，当有一条从
u 到 v 的边时，返回 true

};

bool Reachable(Graph&G, int v, int w){
//请写入你的算法

}
```

6、给定一棵所有边的长度均为整数的树，现要求延长其中某些边，使得从根到任意节点的路径长度相等。问满足要求的树的边长度之和最小是多少？请写出你的算法，并分析时间复杂度。

欢迎回复，给出你的解答。

从没有找工作经历的我今天参加了 [Google](#) 的笔试，本来还自我感觉良好呢，谁知道考题那是嗷嗷不会啊。

考的几乎都是算法，指针特别的多，不过时间太久不用了都忘记了，数据结构的也不少，考了队列，还有一些编译原理的题，关于表达式的；

三道大题第一个还蛮简单，是向双向列表插入一个节点，第二个问题比较恶心，判断 **A** 字符串中的各个字符数目是否不大于 **B** 字符串中的各个字符数目，由于没时间了就没写完，第三题更是相当及其以及特别的恶心，找到整数数组中满足 **A*B=C** 的元素，而且要更优的，算了半天的时间复杂度还是没写出来。

还是忙该忙的事吧，眼看期末考试了，不能挂课，加油！！！

1、有两根不均匀分布的香，香烧完的时间是一个小时，你能用什么方法来确定一段 15 分钟的时间？

答：2 根香同时点燃，第一根两头都点燃，第二根只点一头，第一根点完的时候是半个小时，接着把第二根两头都点燃，第二根点完的时候就是 15 分钟。

2、一个经理有三个女儿，三个女儿的年龄加起来等于 13，三个女儿的年龄乘起来等于经理自己的年龄，有一个下属已知道经理的年龄，但仍不能确定经理三个女儿的年龄，这时经理说只有一个女儿的头发是黑的，然后这个下属就知道了经理三个女儿的年龄。请问三个女儿的年龄分别是多少？为什么？

答：2, 2, 9, 1 岁不可能

3、有三个人去住旅馆，住三间房，每一间房 \$10 元，于是他们一共付给老板 \$30，第二天，老板觉得三间房只需要 \$25 元就够了于是叫小弟退回 \$5 给三位客人，谁知小弟贪心，只退回每人 \$1，自己偷偷拿了 \$2，这样一来便等于那三位客人每人各花了九元，于是三个人一共花了 \$27，再加上小弟独吞了不 \$2，总共是 \$29。可是当初他们三个人一共付出 \$30 那么还有 \$1 呢？

答：没错，三个人付了 27 块，老板拿了 25 块，小弟拿了 2 块

4、有两位盲人，他们都各自买了两对黑袜和两对白袜，八对袜子的布质、大小完全相同，而每对袜子都有一张商标纸连着。两位盲人不小心将八对袜子混在一起。他们每人怎样才能取回黑袜和白袜各两对呢？

答：不知道，还要仔细想想

5、有一辆火车以每小时 15 公里的速度离开洛杉矶直奔纽约，另一辆火车以每小时 20 公里的速度从纽约开往洛杉矶。如果有一只鸟，以 30 公里每小时的速度和两辆火车同时启动，从洛杉矶出发，碰到另一辆车后返回，依次在两辆火车来回飞行，直到两辆火车相遇，请问，这只小鸟飞行了多长距离？

答：记好两车相遇时间，就是鸟飞行时间，乘以其飞行速度就得到飞行距离。

6、你有两个罐子，50 个红色弹球，50 个蓝色弹球，随机选出一个罐子，随机选取一个弹球放入罐子，怎么给红色弹球最大的选中机会？在你的计划中，得到红球的准确几率是多少？

答：不知道，还要仔细想想

7、你有四个装药丸的罐子，每个药丸都有一定的重量，被污染的药丸是没有被污染的重量 + 1. 只称量一次，如何判断哪个罐子的药被污染了？

答：不知道，还要仔细想想

8、你有一桶果冻，其中有黄色，绿色，红色三种，闭上眼睛，抓取两个同种颜色的果冻。抓取多少个就可以确定你肯定有两个同一颜色的果冻？

答：4

9、对一批编号为 1~100，全部开关朝上(开)的灯进行以下操作：凡是 1 的倍

数反方向拨一次开关；**2** 的倍数反方向又拨一次开关；**3** 的倍数反方向又拨一次开关……问：最后为关熄状态的灯的编号。

答：不知道，还要仔细想想

1 0、想象你在镜子前，请问，为什么镜子中的影像可以颠倒左右，却不能颠倒上下？

答：人的眼睛是左右对称的

1 1、一群人开舞会，每人头上都戴着一顶帽子。帽子只有黑白两种，黑的至少有一顶。每个人都能看到其它人帽子的颜色，却看不到自己的。主持人先让大家看看别人头上戴的是什么帽子，然后关灯，如果有人认为自己戴的是黑帽子，就打自己一个耳光。第一次关灯，没有声音。于是再开灯，大家再看一遍，关灯时仍然鸦雀无声。一直到第三次关灯，才有劈劈啪啪打耳光的声音响起。问有多少人戴着黑帽子？

答：**3**

1 超级失败的**1**：说**8** 点开始，考试时间**100** 分钟，怎么算都是**9: 10** 交卷；**9** 点一到匆匆交卷了，晚上躺在床上才发现错也；

2 超级失败的**2**：把自个的生日又记错了；

3 怕怕的发现：发现 **mm** 还是超级可怕滴，眼睁睁看着一个骗局，哎，也得谨慎些以防上当受骗啊；

题目如下：

$T(0) = 1$; $T(1)=1$; $T(2)=2$; $T(n)=T(n-1)+T(n-2)+T(n-3)$;

用最优方式求 $T(n)$;

```
int?T(int?n)?{
}
```

可以用最熟悉的语言写

在考场的第一个做法

```
?1 public ? class ?T? {
?2  ? public ? int ?t( int ?n) {
?3  ?? if ?(n? == ? 0 )? {
?4 ??? return ? 1 ;
?5  ??} ? else ? if ?(n? == ? 1 )? {
?6 ??? return ? 1 ;
?7  ??} ? else ? if ?(n? == ? 2 )? {
?8 ??? return ? 2 ;
?9  ??} ? else ? {
10 ??? return ?t(n - 1 )? + ?t(n - 2 )? + ?t(n - 3 );
11 ??} ?
12 ?
13 }
```

当时发现时间够用，进行了公式推理，但未得出规律的真谛

每个都与 $T(3)$ 可以发生直接关系，关系是 2 的幂次方，但最终没有得出公式

遂改进如下：

```
?1 public ? class ?T? {
?2  ? public ? int ?t( int ?n) {
?3  ?? if ?(n? == ? 0 )? {
?4 ??? return ? 1 ;
?5  ??} ? else ? if ?(n? == ? 1 )? {
?6 ??? return ? 1 ;
?7  ??} ? else ? if ?(n? == ? 2 )? {
?8 ??? return ? 2 ;
?9  ??} ? else ? {
10 ??? return ? 2 ? * ?t(n - 1 )? - ?t(n - 3 );
```

```
11 ??} ?  
12 ?}  
13 }
```

晚上躺在床上，怎么可能这样直接呢？

突然想到最起码的一点就是重复数的计算，应该进行保存；

如果正向逐个求然后保存，可行；

如果倒向如何保存，尚未想好

大家来仁者见仁一下哦（有更好的思路的请指点）

```
public class T {  
?Map values = new HashMap();  
?  
?public int t(int n){  
??int result = 0;  
??if (n == 0) {  
??? result = 1;  
??} else if (n == 1) {  
???result = 1;  
??} else if (n == 2) {  
???result = 2;  
??} else {  
???result =? 2 * t(n-1) - t(n-3);  
??}  
??return result;  
?  
}  
.....
```

一、单选

1、80x86 中，十进制数-3 用 16 位二进制数表示为？

2、假定符号-、*、\$分别代表减法、乘法和指数运算，且

1)三个运算符优先级顺序是： -最高， *其次， \$最低；

2) 运算符运算时为左结合。请计算 $3-2*4\$1*2\3 的值：

- (A)4096, (B)-61, (C)64, (D)-80, (E)512

3、下列伪代码中，参数是引用传递，结果是？

```
calc(double p, double q, double r){q=q-1.0;r=r+p}
main(){
```

```
    double a = 2.5, b = 9.0;
    calc(b-a, a, a);
    print(a);
}
```

- (A)1.5 (B)2.5 (C)10.5 (D)8 (E)6.5

4、求输出结果：

```
int foo(int x, int y){
    if(x <=0 || y <= 0) return 1;
    return 3 * foo(x - 1, y / 2);
}
printf("%d\n", foo(3, 5));
```

- (A)81 (B)27 (C)9 (D)3 (E)1

5、下列哪个数据结构在优先队列中被最广泛使用？

- (A)堆 (B)数组 (C)双向链表 (D)图 (E)向量

6、以下算法描述了一个在 n 个元素的双向链表中找到第 k 个元素的方法 $k \geq 1$ 且 $k \leq n$ ：

如果 $k \leq n - k$, 从链表开始往前进 $k-1$ 个元素。

否则, 从终点出发, 往回走 $n - k$ 个元素。

这个算法的时间代价是？

- (A) $\theta(n\log n)$ (B) $\theta(\max\{k, n - k\})$ (C) $\theta(k + (n - k))$
(D) $\theta(\max\{k, n - k\})$ (E) $\theta(\min\{k, n - k\})$

7、有一个由 10 个顶点组成的图，每个顶点有 6 度，那么这个图有几条边？

- (A)60 (B)30 (C)20 (D)80 (E)90

8、正则表达式 $L = x^*(x|yx^+)$ 。下列哪个字符串不符合 L

- (A)x (B)xyxyx (C)xyx (D)yxx (E)yx

9、为读取一块数据而准备磁盘驱动器的总时间包括

- (A)等待时间 (B)寻道时间 (C)传输时间 (D)等待时间加寻道时间
(E)等待时间加寻道时间加传输时间

二、算法

1、打印出一个二叉树的内容。

2、在一个字符串中找到第一个只出现一次的字符。如 abaccdeff，输出 b。

3、给定一个长度为 N 的整数数组（元素有正有负），求所有元素之和，最大的一个子数组。分析算法时空复杂度。不必写代码。

附上动态规划做法的答案：

最大子序列

问题：

给定一整数序列 A_1, A_2, \dots, A_n (可能有负数)，求 $A_1 \sim A_n$ 的一个子序列

$A_i \sim A_j$ ，使得 A_i 到 A_j 的和最大

例如：整数序列 -2, 11, -4, 13, -5, 2, -5, -3, 12, -9 的最大子序列的和为 21。

对于这个问题，最简单也是最容易想到的那就是穷举所有子序列的方法。利

用三重循环，依次求出所有子序列的和然后取最大的那个。当然算法复杂度会达到 $O(n^3)$ 。显然这种方法不是最优的，下面给出一个算法复杂度为 $O(n)$ 的线性算法实现，算法的来源于 **Programming Pearls** 一书。在给出线性算法之前，先来看一个对穷举算法进行优化的算法，它的算法复杂度为 $O(n^2)$ 。其实这个算法只是对穷举算法稍微做了一些修改：其实子序列的和我们并不需要每次都重新计算一遍。假设 $\text{Sum}(i, j)$ 是 $A[i] \dots A[j]$ 的和，那么 $\text{Sum}(i, j+1) = \text{Sum}(i, j) + A[j+1]$ 。利用这一个递推，我们就可以得到下面这个算法：

```
int max_sub(int a[], int size)
{
    int i, j, v, max=a[0];
    for(i=0; i<size; i++)
    {
        v=0;
        for(j=i; j<size; j++)
        {
            v=v+a[j];//Sum(i, j+1) = Sum(i, j) + A[j+1]
            if(v>max)
                max=v;
        }
    }
    return max;
}
```

那怎样才能达到线性复杂度呢？这里运用动态规划的思想。先看一下源代码实现：

```
int max_sub2(int a[], int size)
{
    int i, max=0, temp_sum=0;
    for(i=0; i<size; i++)
    {
        temp_sum+=a[i];
        if(temp_sum>max)
```

```
    max=temp_sum;
else if(temp_sum<0)
    temp_sum=0;
}
return max;
}
```

在这一遍扫描数组当中，从左到右记录当前子序列的和 `temp_sum`，若这个和不断增加，那么最大子序列的和 `max` 也不断增加(不断更新 `max`)。如果往前扫描中遇到负数，那么当前子序列的和将会减小。此时 `temp_sum` 将会小于 `max`，当然 `max` 也就不更新。如果 `temp_sum` 降到 0 时，说明前面已经扫描的那一段就可以抛弃了，这时将 `temp_sum` 置为 0。然后，`temp_sum` 将从后面开始将这个字段进行分析，若有比当前 `max` 大的字段，继续更新 `max`。这样一趟扫描结果也就出来了。

[google](#) 的魅力就不用词语形容了,从晚上的人气就可见一斑.准点赶到教七,却发现门口已经被人群给堵了,[google](#) 的工作人员一个劲地劝大家别进去了,理由是"为了安全":).也好,不用听开复同学的唠叨.回教三的路上,一打打的人迎面而来,熟悉的,陌生的,本校的,外校的.我彷彿到了麦加,穿梭在朝圣的人流中.但是念佛的人多,成佛的却是寥寥.晚上肯定有四位数的申请者,不过能有多少人登上幸运的快车呢?能突破个位数吗?所谓千军万马过独木桥,找工作,挺残酷!

于我,去参加笔试,更多是一种体验.选择 [google](#),权当是我对这家伟大的公司的一点敬意吧!

在软件上,我这样的水平,恐怕连菜鸟也谈不上.那么硬件呢?光学呢?我的竞争力在哪里?不能不说,对于找工作,我是有所恐惧的.现在的选择,也许不过是种逃避.

昨天,zhengzheng 说他改变主意,准备工作了,一起走的兄弟,又少了一个.今天,simon 说在犹豫是否调整申请的方向,因为他想以后还是进公司的.说起,5 年的努力是否值得?未来也许只有上帝知道吧!

乘着脑子还清晰,先把笔试题记下:

9 个选择题,2 个编程,1 个算法.选择题,感觉比较基础,不知里面有没有设地雷?反正没啥印象了.

编程 1,打印二叉树,实现语言不限,先后也不限.

编程 2,给定一个字符串数组,从中找出第一个只出现一次的字母.

算法题,给定一个整数数组,从中切出一个连续片段,保证其元素和最大.求最优算法,分析时间和空间复杂度.