1.开放式问题,有些网站每天只允许有限次访问,怎么抓取网页使得索引尽量全面和新鲜  
？？  
2.在C++文件中只declare class A, 但不以任何方式define class A, 是做什么用  
3. Estimate the time cost of transfering 1M of data from one memory stick to  
another.  
        - when the data in memory is sequentially stored;         - when the  
data in memory is stored in blocks;         - does the bus width matter   
here?  
4.How to transform a unbalanced tree into balanced tree?  
第2个题我想的是保留A的名字，以后再定义  
第四个题我想的是先算每个节点的blance factor然后再调整，具体怎么调整就不知道了。  
第四个题还想到一个办法是转成双链表，然后再转balanced tree，保证了inplace

Phone:  
/////////////////////////////////////////////////////////////////////////////////////////////////////////////  
SSL, for example   a -> b (A refer B)  
                   d -> c (D refer C)  
                   c -> a (C refer A)  
find a data structure to save these information and print the following   
result:  d->c->a->b

The data structure I pick is Array Hash Table.   
  
Typedef struct NODE{  
    unsigned char keyVal;  
    unsigned char \*pRefer; // point to the char referred.  
} Node;  
  
Node key[26];  
  
Insertion operation: space O(26), time O(1)  
  
int insertPair(unsigned char refer, unsigned char \*referred)  
{  
     key[refer-'a'].pRefer = referred;  
}  
  
Print function:  space O(26), time O(n)  
  
we need a auxiliary list to sort the linked list.

Onsite:  
1. design public interfaces for cache (like cache for string)  
2. design public interfaces for router  
3. merge two binary search tree

终极方案：O(logn + logm)的复杂度，O(1)的空间，把BST a和b互相挂。  
  
public BST merge(BST a, BST b) {  
     if (b == null) return a;  
     if (a == null) return b;  
  
     if (a.root < b.root)   
         return merge(b, a);  
  
     // now, a.root >= b.root  
  
     BST bleft = b.left;  
     BST bright = b; bright.left = null;  
  
     // we know every node in bleft is less than a.root  
     if (a.left == null) {  
         a.left = b;  
     } else {   
         a.left = merge(b, a.left);  
     }  
  
     a.right = merge(a.right, bright);  
  
     return a;  
}

4. 爬楼梯，每次可以一个step，或者两个step，问有多少种走法？  
5. 如何实现一个queue?　怎么分别通过stack,linked list, heap实现？  
6. preorder　binary search tree.  
   判断两个binary search tree的preorder序列相同不相同

可否把空间节省点？到O(1)。  
  
写2个iterator。  
  
public boolean hasSamePreOrder(BST a, BST b) {  
     Iterator ait = a.iterator();  
     Iterator bit = b.iterator();  
  
     while (ait.hasNext() && bit.hasNext()) {  
          if (ait.next() != bit.next())  
               return false;  
     }  
  
     return ait.hasNext() && bit.hasNext();  
}  
  
问题就转化成了使用空间O(1)，实现iterator。  
7. 判断两个树相等不相等

DFS遍历，判断是否相等咯。  
  
public boolean equals(BST a, BST b) {  
     if (a == null) return b == null;  
     if (b == null) return a == null;  
  
     return a.value = b.value  
                && equals(a.left, b.left)   
                && equals(b.left, b.right);  
}  
8. 判断两个DAG(有向无环图)相等不相等

DFS遍历，判断是否相等咯。  
  
public boolean equals(DAG a, DAG b) {  
     if (a == null) return b == null;  
     if (b == null) return a == null;  
  
     if (a.value != b.value)  
          return false;  
  
     DAG[] anext = a.child();  
     DAG[] bnext = b.child();  
  
     if (anext.length != bnext.length) {  
          return false;          
     }  
  
     for (int i = 0; i < anext.length; i++)  
          if (!equals(anext[i], bnext[i]))  
               return false;  
  
      return true;  
}

1. 算时针和分针的夹角  
----------------------------------------------------  
  
整个钟面是360度，算算时针的位置和分针的位置，然后求差的绝对值即可。  
  
double foo(int hour, int minute) {  
    assert (minute < 60 && minute >= 0);  
    assert (hour < 24 && hour >= 0);  
  
    if (hour >= 12)   
         hour -= 12;  
  
    double minuteAngle = 360. \* minute / 60  
    double hourAngle =   
            360. \* (hour / 12)  +   
            minute / 60. \* (360 / 12);  
  
    return Math.abs(hourAngle - minuteAngle);   
}  
  
----------------------------------------------------  
  
3. 判断List有没有环，分析时间复杂度  
  
----------------------------------------------------  
  
用2个指针，一个快的，一个慢的。若是2个指针重叠了，那么就是有环。  
  
public boolean hasLoop(Node head) {  
    assert (head != null);  
  
    Node p1 = head;  
    Node p2 = head;  
  
    while (true) {  
         if ((p1 = p1.next) == null)  
              return false;  
         if ((p1 = p1.next) == null)  
              return false;  
         if ((p2 = p2.next) == p1)  
              return true;  
    }  
    return false;  
}  
  
时间复杂度是O(n).

1 compare binary search tree & hashtable  
2 design a hashtable to print words between 'B' and 'C'  
3 design a general hash function for words(like words in dictionary)  
4 why multh-threading? implementation in C/C++  
5 mutually exclusive, disadvantage of mutual exclusion?  
6 describe your best project

早上10点半开始。面第一位阿三哥，开始侃项目，谈跟专业相关的东西，追问的很深，  
最后还有15分钟的时候要求写代码，要求inplace对一个数据结构内的元素重新排序，  
昏倒啊。在白板上画了简单的结构，讨论后获得首肯，然后开始写代码。  
  
我把笔记本电脑带了过去，所以在键盘上敲。大概一会就写出来了。（若是在白板上写  
，怎么死的都不知道阿）。然后三哥问，are you done？我说跑几个测试案例试试看。  
然后在纸上写了5个案例，一行一行的检查。立马发现2个bug，更正之。三哥看到快没  
时间了，说你跑这个案例试试。遂发现一个新的bug，更正之。最后代码写出来完整。  
三哥满意的走了。  
  
第二位是个白人。一上来就做题。开始我理解以为是一个DP，后来沟通之后发觉可以有  
很简单的解法。直接奔向O(n)的解法。跟面官沟通完想法，最后获得同意后在笔记本上  
敲键盘。很快写了出来。面官随后问测试案例。我直接在笔记本电脑里面写测试代码，  
写了10多个。面官随后表示满意。我表示可以compile跑跑看。解决了compile的error  
，有一个测试案例不通过。解决bug，最后所有测试案例通过。  
  
下午，第三位是个黑人。这场面试最凄惨了，希望我的经验教训能帮助大家。开始一个  
很简单的题目，2叉树遍历的，5分钟不到搞定，写了代码。然后黑人老兄把问题延伸，  
问了一个复杂的情况，我没有跟面官沟通，就直接写代码。有错误。被指正出来，再修  
改代码，面官说还是有错误，在修改代码，面官说还有错误。最后我说，我们得沟通沟  
通，遂又回白板开始讨论算法。最后一刻把问题解决出来，但是没时间写代码了。我看  
了下黑人老兄的笔记本电脑，昏倒，他把我的每一个版本的代码全部写了下来！包括第  
一个版本有错误的，然后第二个版本，第三个版本。遂后悔不已，应该先在白板前沟通  
好再写代码的。当时黑人老兄在我说完算法之后面无表情，我应该询问是否有无bug。  
不过最后黑人老兄说，虽然你没有时间写正确的代码了，但是能走到这步能把问题解出  
来的人不多。  
  
第四位一个白人，一进来沉着脸，一副别人欠你钱的表情。开始一个简单的问题，  
Programming Pearl上第一章的案例题，大家都知道了吧。之后遂把数据规模提高到  
10^10。我给了2个解法。之后把数据规模提高到10^15，输入已经无法存在一台电脑  
上，说咱们有1000个电脑，每个电脑上存一部分输入，你怎么解决。这个磨蹭了好久，  
最开始我有个brute force的想法，当时没有说出来，觉得不够好，后来跟面官折腾了  
半天，才发觉最开始的想法就是他想要的。最后白人老兄终于脸开笑容，握手拜拜。  
  
第五位一位三哥，讨论open end question。但是问的很深，涉及到数据库的实现，多  
线程，cache的实现，Javascript，等等等等，这些都是我简历上写做过的东西，我把所  
有可能的情况全部都列了出来，说的嗓子都哑了。三哥面无表情，但是自我感觉还说的  
挺多挺全的

Write the clone method of a linked list whose one node point to some random node.

You are given the amazon.com database which consists of names of millions of products. When a user enters a search query for particular object with the keyword say "foo" , output all the products which have names having 50% or more similarity with the given keyword ie "foo" Write the most efficient algorithm for the same.

[http://www.careercup.com/question?id=3408663](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D3408663&sa=D&sntz=1&usg=AFQjCNEVlDdltAnXGHOrRpPZNhFO4do7Tg)

You are given a dictionary of all valid words. You have the following 3 operations permitted on a word: delete a character, insert a character, replace a character. Now given two words - word1 and word2 - find the minimum number of steps required to convert word1 to word2. (one operation counts as 1 step.)

Given an array of n elements and an integer k where k<n.  
Elements {a[0].....a[k] and a[k+1].....a[n] are already sorted. Give an algorithm to sort in O(n) time and O(1) space.

en n number of points in space and a point P, find 3 nearest point to p

[http://algorithm.chaoskey.com/02/02](http://www.google.com/url?q=http%3A%2F%2Falgorithm.chaoskey.com%2F02%2F02&sa=D&sntz=1&usg=AFQjCNHpjTpGMfxSyFjkcJ5KAUh8igbh4w) 大整数乘法

一面大概1小时 用google doc写java code  
1. 最challenging的project 问的很细 关注challeing在哪 怎么解决的  
2. abstract class和interface的区别 什么时候用哪个  
3. 实现List<PhoneNumber> deduplicate(List<PhoneNumber> phoneNumbers) {}   
我先写把list加到一个set里面然后把set包装成list出来 他就笑了说不给这么搞 用别的data   
structure 然后我就写了个用HashMap的.然后问复杂度 然后问hashCode怎么写 其实后来想想  
用HashMap的话和原来是一样的 都靠的是HashMap的keySet是一个set  
4. reservoir sampling. 实现List<Node> getRandomSample(Iterator<Node> itr,   
int sampleSize) {} 返回sampleSize个随机的元素 因为只给了Iterator拿不到  
collection的size我就说入过给的是collection,那么有size,有sampleSize可以算下  
possibility看iterate的时候当前元素要不要加到result list里面去 只想到了这么多  
后来搜了下 贴个解  
http://eyalsch.wordpress.com/2010/04/01/random-sample/  
  
  
二面是老印 一听到心里就一凉 唉 都有心理阴影了  
1.问了几个inner class的情形, static和non static的区别各什么时候用,anonymous  
什么时候用  
2.interface和abstract class的区别  
3.garbage collection原理是怎样的  
然后就写code了 全是bst, insert, count nodes, count nodes non-recursive,   
isBST 最后isBST我还是写错了。。。他一说我就知道挂了 唉 move on吧

开始coding, 很简单，关于字符串的，不过没听清楚他在问什么，接着他讲了个  
例子才明白。然后，simple solution，O(n^2), 接着binary search 优化O(nlgn),接  
着bit map 优化 O(n),

1. 写程序判断一棵二叉树是不是对称的。  
   2.写程序求一个词到另一个词的最短变换路径。（二词长度相等）  
   3.Design auction ads bid的data holder class  
   4.Design Amazon 主页服务器的auto recovery，load balancing，应该minimize的指  
   标，针对指标估计标准值；对含有user ID信息的request如何load balancing。  
   5.提出一个对Amazon有用的feature，design。  
   6.假如你是一个start up公司的tech management最高层，公司的网站最近有些慢，如  
   何分析解决。（假设面试官是CEO)  
   7.写程序分层打印一个矩阵，如下例：  
   123  
   456  
   789  
   打印输出的顺序是：123698745  
   8.Machine learning的一些概念，bias, variance, boosting,SVM&DecisionTree&  
   NeuralNetwork比较  
   9.design 算法，抽取不同product record中对应同一属性的不同值，比如  
   Nike Shoes, Black, Size 7, ID: WSLT328764  
   Nike Shoes, White, Size 9, ID: WSLT239043  
   Black White, Size 7, 9都是属性  
   10. 一个巨大的File (billions of rows)，每行包含两个field：ID, description。  
   设计算法，找出所有对应duplicated description的ID。  
   11. 介绍以往的project，个人的working style,etc.

第一个是老美，先问了一些简单问题，比如怎么判断一个32 bit是big endian 还是  
small endian等等。最后出了一道算法题，也很容易，给定K个sorted array，要求输  
出一个大的sorted array。简单的merge sort就解决了。不过merge sort 要求每次K个  
array中，最小的element。简单的当然是scan这K个array。我提出可以把K个array的当  
前element放入Heap structure，这样每次搜索就从O(K)降低到O(logK)。最后写了个程  
序。  
  
第二个是老中。也是先问了一些简单问题，然后让我设计一个分布式文件系统，给定  
path name，可以读写文件。具体的system design这里就不提了。其中一个细节是，给  
定path name，怎么知道哪个node拥有这个文件。我提出需要实现一个lookup function  
，它可以是一个hash function，也可以是一个lookup table。如果是lookup table，  
为了让所有client sync，可以考虑额外做一个lookup cluster。然后Interviewer很纠  
结，既然可以用hash function，为什么还搞得那么复杂。我就告诉他hash function的  
缺点。假定一开始有N个node，hash function把M个文件uniformly distribute到N个  
node上。某天发现capacity不够，加了一个node。首先，要通知所有的client machine  
，configuration 改变了。如果不想重启client machine的process，这不是一个  
trivial job。其次，文件到node的mapping也变了。比如，本来按照hash function，  
一个文件是放在node 1。加了一个node 后，它可能就map到node 2了。平均来说，N/(N  
+1)的文件需要move到新的node。这个data migration还是很大的。然后我就提出一些  
hash function的design，可以减少data migration。  
  
最后他提了一个问题，说要实现一个function，要统计distributed file system所有  
目录的大小。前提是，一个目录下的文件可能放在不同的node上。我说这个不就是在每  
个node上统计，然后发到一个merge吗。他说对，但是又问用什么data structure来表  
示。我说这就是hash table，key就是directory name，value就是大小。因为  
directory本身是树结构，这个hash table的key可以用tree来组织。最后让我实现一个  
function，把我说得这个data structure serialize成byte array。因为这个byte   
array就是网络传输的data。我用了depth first traverse。不过等我程序写完，才发  
现，用breath first traverse会更方便，code也会很简洁。  
  
第三个也是老中。他可能没有很好的准备，问题一开始有点含混不清。花了一点时间，  
基本明确，他是要我用pthread实现thread pool，以及thread job management。先是  
define class interface，然后用pthread的mutex和semaphore实现了consumer/  
producer queue。这个queue允许users（producers)加入thread jobs，thread   
managers(consumers)拿出thread jobs，并执行。当场design class interface，并做  
到面面俱到有点难，好在  
我山寨了Java的thread class。有了interface，implementation还是比较容易的。他  
顺便也问了一些multiple thread的问题，比如怎么做singleton等等。  
  
第四个是老印。他问了一道算法题，假定有个graph，怎么找出不带circle的最长path  
。我纠结了很久，最后用dynamic programming 解决的。好在他主要focus在idea上面  
，没让我把code写完。等我想清楚算法，一半时间已经过去了。要写完code，我还真做  
不到。

电话面试：1个小时  
4个面试官， 主要问工作经验, 多线程，socket，程序优化相关  
  
然后第一次onsite：3个小时  
2个VP, 4个AVP  
项目相关的问得很多，基本上差不多整个项目实现的细节都问了， 包括多线程模型，  
memory pool实现， socket 模型（select和async socket实现）， 异步文件读写，   
内部使用的数据结构， 和IPC 实现， 跨平台的实现方法(thread， socket, timer,   
TLS, fast mutex实现）  
coding： 比较简单， 就是c的字符串操作  
  
第二次onsite: 1个小时  
1个VP, 1个AVP  
C++/template， JAVA, C#，   
估算某个building 每个月的电费。。。这个很晕  
2个coding: C字符串操作

Sorting  
    o    Bubble/select/insertion/counting/qsort/heap/merge/bst  
    o    Time/space complexity analysis  
  
•    Caching design  
    o    Replacement policy (LRU, LFU, NRU, etc…)  
    o    Efficiency/complexity/performance  
    o    Distributed cache  
    o    Hashing  
  
•    Multi-thread  
    o    Locking/mutex/semaphore/critical section  
    o    Coding pattern  
            producer/consumer (aka writer/reader)  
            ref-counting  
  
•    Tree  
    o    Heap  
    o    BST/black-red (BR/AVL is a stretch)  
    o    B+  
    o    Suffice/prefix trees (trie)  
    o    Expression tree  
  
•    DP (dynamic programming)  
  
•    Search  
    o    Basic backtrack  
    o    Backtrack trimming  
    o    Breadth first vs depth first search  
    o    A\*  
    o    Bidirectional search  
  
•    Graph  
    o    Shortest distance (djstra, floydwarshall)  
    o    Cycle detection  
    o    Flow network algorithms  
  
•    List  
    o    Stack/queue/priority queue  
    o    List manipulation  
  
•    Pattern matching  
    o    Strict string matching  
    o    Wildcard matching  
  
•    Bit operations  
  
•    Compression algorithm.  
    o    Huffman  
    o    RLE (run-length encoding)  
    o    LZW  
  
•    Design  
    o    Design pattern:  singleton, factory, provider, witness, command,   
etc  
    o    OOP designs  
            Interface vs abstract class  
            Virtual behaviors  
    o    Design a feature like:  
            Twitter  
            Facebook/linkedin friend recommendation  
  
•    C#/Java  
    o    GC algorithm  
  
•    Tech  
    o    Hadoop  
    o    Mapreduce  
    o    TSQL/SQL  
    o    Memcached  
    o    Membase  
    o    Cassendra  
  
•    Reason to use/not to use STL/ATL  
  
•    Exception versus error code.  
  
•    Memory allocator/heap manager  
  
====================  Actual coding/interview questions encountered lately =  
===================  
主要面试的对象集中在high scale/high perf/highly available/distributed   
service infra areas. 所以题目有点偏..  
  
仅供参考，并不准备一道一道解释回答。Sorry.  
  
==============================================  
1.    Say a compare function f  
F(x, y) returns 1, if x is better than y  
F(x, y) returns 0, if x is not better than y (but doesn’t say x is worse   
than y, it’s simply saying x isn’t better than y)  
F is communitive: F(x, y) -> 1 then F(y, x) -> 0  
But F isn’t transitive: F(x, y) -> 1 && F(y, z) -> 1, doesn’t mean F(x, z)  
-> 1  
  
So based on the rules above, write an efficient algorithm to find an element  
in a list that is the best. If x is the best, it means for any y (that y !=  
x) in the list, F(x, y) -> 1. The function can return NULL to indicate   
there isn’t such an element exist.  
  
  
==============================================  
2.    Two strings (ASCII, value range is 0 – 255), test, and alpha. Write a  
function to return true if every character in test appears in alpha, and   
return false otherwise.  
  
  
==============================================  
3.    Say we have a simple file system on disk is like the following:  
  
There are many consecutive sectors on disk, the sector 0 is the file index   
table. Each of the following sector would either belong a file, or empty.   
Each sector has a head structure, which contains info 1) whether this is an   
empty sector or not and 2) if this is not an empty sector, so it belongs to   
a file, then what’s the next sector in the file. If this field is null,   
then this sector is the last sector for this particular file.  
  
The file index table contains a list of entries, each has a file name, and   
its first sector number.  
  
Now somehow the sector 0 (file index table) is completely corrupted, write   
an efficient algorithm to rebuild the file index table. Filenames can be   
generated randomly as long as they’re unique.  
  
  
==============================================  
4.    Say have a N computer, each computer gets assigned with a random   
integer value. We need to write a single program, that the same program will  
be run on all the N computers at the same time. At the end, every program   
on each computer needs to print out the sum of all the integers associated w  
/ all computers.  
  
To achieve this, there are two synchronous blocking APIs to use.  
  
Send(k, n): if a program calls this API, it means it wants to send an   
arbitrary integer n to computer k (0 <= k < N).  
Receive(k): if a program on computer j calls this API, it means it wants to   
receive the integer from computer k where the program sends that integer  to  
computer j.  
  
Both API are sync/blocking. That means if there is sender, but on the   
receiving end, there is no receiver, then the sender will be blocked and   
wait until corresponding receiving API is called on the target computer.   
Vice versa.  
  
Two metrics for achieving this goal. 1) # of msgs (send -> receive counts as  
1 msg) and 2) assume each msg takes time t to complete, the total T.  
  
Try to write the program to 1) achieve the goal and 2) try to optimize T so   
T being the smallest  
  
  
==============================================  
5.    Say you have an array of bytes (value is [0-255]), write an (LRE)   
compression algorithm to offer fast, efficient, on the flight compression   
and decompression.  
  
  
==============================================  
6.    Given an integer array, calculate total number of pairs (x, y) in the   
array, such as x > y.  
For example, given an array of 5, 6, 2, 1  
All the pairs that have descending order are:  
5, 2  
5, 1  
6, 2  
6, 1  
2, 1  
So result = 5.  
  
  
==============================================  
7.    Minesweeper  
a.    Design data structure for the board and its mines  
b.    Given n (board width), m (board height), and k (number of mines),   
efficiently and randomly generate k mines and place them in the board.  
  
  
==============================================  
8.    itoa  
  
  
==============================================  
9.    given an array of integers, find the smallest number; find two   
smallest numbers; find k smallest numbers;  
  
  
==============================================  
10.    F(a, b) -> true or false means that person a knows person b. But it’  
s not commutative, because person a can know person b, but b may not know a.  
Say, you throw a party, you knows a number of people, invite them, then each  
of them knows a bunch of people (those sets of people can intersect), so on  
and so forth.  
Now there is a celebrity comes in, everyone in the party knows him, but he   
knows no one.  
Write an algorithm effectively find out who the celebrity is.  
  
  
==============================================  
11.    Say you’re Walter Disney and you have so many copyrighted videos,   
and you want to write an algorithm (high level description is good enough)   
to detect any video clips on YouTube.com are violating copyrights.  
  
  
==============================================  
12.    Given two identical length strings, say “abcdefg”, “bcdefgx”,   
first define the distance (or think in search term, relevance) between these  
two strings, and then write an algorithm.  
  
  
==============================================  
13.    Master mind guess.  
Say a random 6-digit number is generated and hidden secretly.  
Then there is a function: int Tell(int n); where the n is any 6-digit number  
, and the function Tell returns how many of the digits are correct and in   
the right position.  
For example, if the secret is “123456”, and Tell(“523499”) -> 3  
Now write an algorithm such that you can call Tell to figure out what is the  
secret number efficiently.  
  
  
==============================================  
14.    a stream of integer coming in (in single link list fashion), you don  
’t know how many, but you can use current->next == NULL to know if the   
stream ends.  
  
Now given a K (assume K < count of total number of element in the stream,   
even though you don’t know the total number until you fully scan the stream)  
  
Output randomly sampled K integers from this stream.  
  
(reservoir-sampling)  
  
This is very similar to #7, the random generate minesweeper problem.  
  
  
==============================================  
15.    say two linked list may or may not merge at certain node. If they   
merge, then it will be like a Y shape structure.  
  
Write a function, taking two linked list and 1) return true/false whether   
these two linked lists merge or not and 2) return the node where the merge   
occurs.  
  
  
==============================================  
16.    swap every pair two adjacent nodes in a linked list.  
  
  
==============================================  
17.    Write a function to find the nth last element from a Linked List.  
  
  
==============================================  
18.    Given a linked list, findout wether it is a palindrome or not,  
  
Idea: find mid point, then compare.  
  
  
  
==============================================  
19.  
  
struct node  
{  
  int value;  
  node \*next;  
  node \*random\_ref;  
}  
  
How to write a function to take a link list in such structure and dup it   
into a new link list with both next and random\_ref both honored for each   
node in the new list?  
  
  
==============================================  
20.    given two link lists, determine whether they’re reverse of each   
other.  
  
  
==============================================  
21.    1) merge sort on array 2) merge short on linked list.  
  
  
==============================================  
22.    LCS (longest common subsequence), given two string a, b, and   
calculate their LCS value.  
  
  
==============================================  
23.    edit distance. LCS is in fact a sub set of the edit distance problem   
(aka Levenshtein distance)  
  
Edit distance:  
Two strings, a, b, use minimum number of following operations to make them   
identical  
Insert/delete/sub  
While LCS, the set of operations allowed is insert/delete  
And hamming distance, the set of operations allowed is only sub (thus two   
strings must have the same length)  
  
Dynamic programming.  
  
  
==============================================  
24.    write a function to find mid point of a linked list. two methods 1)   
two scans, and 2) use fast/slow pointers.  
  
  
==============================================  
25.    find the longest palindrome in a string.  
  
Solution1: Brute force  
Solution2: string A, reverse it becomes A’, so the problem changes to find   
the longest common substring . Note, it’s different than #22 below, that   
one is about longest common subsequence. But the idea is the same using DP.  
  
  
==============================================  
26.    Do you know about the recommendation engine built/used by Amazon.com   
? How would you build one ? Now use what you know to build a relevancy   
engine for Bing Search.  
  
Backend module:  
•    Storage: for transactional data; logging; system health  
•    Caching:  
Mid –tier modules:  
•    User module: retrieving user info, user history, user social info  
(friends, etc).  
•    Product/action relevancy: if user click kindle, we should find   
related products such as kindle cover, skins.  
•    Popularity module: for each item, what’s the popularity  
•    Campaign: amazon can have an active campaign promoting kindle, no  
matter what user is doing   
•    Feedback/improvement: once recommendations served, what are the   
user follow-up actions, click through rate, etc, to retrain those numbers in  
those components.  
  
  
==============================================  
27.    Design a logging system for an application server? see to it that   
Logging system you define does not include a large overhead in case of large  
loads to server ?  
•    Lossless or eventual consistency or best effort?  
•    In memory ring buffer for speed (flight blackbox)  
•    Async persistent write  
•    Central collection mechanism such can be throttled and retried   
easily  
  
  
==============================================  
28.    How do you design cache server for a simple web application.  
How do you make sure of the data consistancy.  
How do update your data/cache.  
  
  
==============================================  
29.    Design a Hotel reservation system which will support the following   
functions.  
a) User will get a list of all different types of rooms.  
b) User selects a room type & check the room availabilty between the   
specified dates.   
c) User Makes Reservation.  
[Discussed about "locking" the room availbilty or not in case if user wants   
to proceed with reservation]  
  
  
==============================================  
30.    If you were integrating a feed of end of day stock price information   
(open, high, low, and closing price) for 5,000 companies, how would you do   
it? You are responsible for the development, rollout and ongoing monitoring   
and maintenance of the feed. Describe the different methods you considered   
and why you would recommend your approach. The feed would be delivered once   
per trading day in a comma-separated format via an FTP site. The feed will   
be used by 1000 daily users in a web application  
  
  
==============================================  
31.    Imagine that there are 7 servers running in parallel. What happens   
when you need to expand to 20 live? What are issues? What could you do to   
fix this issue in the future  
  
  
==============================================  
32.    How to implement a LRU cache. Fast item retrieve, fast to kick LRU   
item out, fast to add items in.  
  
Combination of doubly linked list (dll) + hash.  
  
Dll is a sorted list ordering from least to most recently used items.  
Hash store <key, ref> to those items.  
  
Operations need to be supported:  
  
Get-item  
Kickout-item  
Update-item (means it’s been used, this can actually be folded into the get  
-item)  
Insert-item  
Delete-item  
  
Most these operations first go into hash map to find the key, and locate the  
item in the linked list, then do certain operations such as remove, move to  
head, move to tail, insert, etc.  
  
More about caching strategies: http://faq.javaranch.com/java/CachingStrategies  
  
  
==============================================  
33.    say char a-z maps to 1…26, A-Z maps to 27…52  
  
Give you a numeric string sequence, such as "123", there would be multiple   
possible mappings: "abc", "lc", "aw". But for "101", there is only one   
possible mapping: "ja". And for "00", no possible mappings.  
  
Write a function, given a numeric sequence in string, return the total   
number of possible mappings.  
  
  
  
==============================================  
34.    string/pattern matching. pattern can contain a-z and special chars   
like "." and "\*"  
  
"." means matching any single char  
"\*" means matching zero or more chars of any kind.  
  
string can contain a-z and "." "\*". Note "." and "\*" in string are just   
regular chars, no special meanings.

题1: UITableView＋NSArray，白板写代码  
题2: NSDictionary，直接在电脑上写代码  
题3: 动态规划智力题  
题4: Objective-C与C++的比较，优劣。

先让我写  
N!我写了递归，然后又让用非递归写了一次。继续问递归的确定。接着问求fib数怎么  
写代码，这些  
代码早练过了，所以不是问题。本来想给他show下我logN的算法，后来他没要求就不写  
了。还问了  
些stack里面存了哪些东西，以及顺序，顺序是和编译器有关。其他问题我也忘记了，  
反正这个哥们  
是物理的PHD，问到问题和数学有点关系。  
然后就是2个以后的同事来面我，问的题目还有点水平。先是OOP的设计，关于公司的船  
运货物到不同  
港口，怎么设计这个系统。具体是怎么样的，我也忘记了。接下来就是关于这个OOP的  
算法设计，问  
如何计算一个港口哪段时间船最多，给你每搜船进出港口的时间。这个题和facebook的  
一个puzzle  
如出一辙。接下来是写了一段代码让我找错，这个很简单，常见错误。还问了我些设计  
模式的题目，  
问我用过哪些，怎么用的。  
第二天是周五，晚上猎头就说feedback好，问我要求多少钱。周六周日他家VP就跟我商  
量offer  
了，其实我根本不想这么早就商量，microstrategy onsite在下周呢。不过他们特别着  
急，所以  
没办法，就接了。 offer是85k 2000股票 无bonus。  
  
周3又去面了microstrategy。  
电面是一个中国人面的，本来hr说面题都是brain teaser，结果问题完全是围绕我简历  
，问的非常  
细致，具体忘记了，最后一个简单的brain teaser  
onsite 一共4个人  
先做题1小时，本来是1个半小时的题，为啥就让我做1小时。。。。  
题目不难，除了一个题很傻，条件不全让无知群众受害。给你level order的序列，让  
你重建BST.  
还有一个LCS  
然后就是进来2个小兵，看着就不是那种很聪明的人，果然问的问题都很简单，害的我  
还故意给他不好  
的方法，然后给他点让我改进的机会。比如大小是100的数组，数的范围是1-99，有一  
个重复，怎么  
找出重复的。我给的方案很多，基本版面有的都给了。然后联系到排序问了我2个排序  
复杂度。  
接下去就是brain tease，设计模式的问题，都不难。最后还有8分钟就把题目问完了，  
只能让我问  
他们问题。我就发挥了下，问了他们很多。  
接着是个manager，问了个比较新的问题，但是办法很老。关于树的题目，就用递归搞  
定。还问了我  
一个OOP的设计问题，主要考我数据结构。反正我给他的方案他满意。  
到吃饭时间了，于是一个美国人带我吃饭。天下没有免费的午餐，果然是啊。路上先问  
我做过的  
project，又到我发挥了。说过n次的了，当然不是问题。不过他还挺懂的，我说的信号  
处理的东西  
他也能明白一些。接着该他发彪了，智力问题是开胃菜，然后就是线程同步问题，如何  
解决死锁。还  
有一个OOP设计问题，设计一个高速收费站，里面有bar，投币机器，地下还有一个秤，  
判断是不是  
卡车。我给的方案里面用到观察模式。老美的智力题也太老了，问来问去就那些。最后  
我说我喜欢当  
面试官，可以出题，多好。  
吃完饭，我以为还有几轮，结果就是最后一轮了，进来一个老印，给我个名片。我居然  
没看.....我  
知道他有点来头，表情就没其他人那么和蔼。问我为啥EE的找CS工作，我说我擅长写软  
件，而且算  
法，数据结构是强项。以前读EE完全是因为对图像处理有兴趣，然后被忽悠到匹大。结  
果来了发现没  
有老师有项目。接着给我出题，就是一个查黄页的题，问我300万个名字，一页有1000  
个名字的黄  
页，要找到一个人的电话，需要查几次。太简单了，binary search，log1500。 然后  
问大概是  
多少。  
问完问我有什么问题， 我说他们的开始的test里面条件不全，希望他们能改了。还有  
如果有offer  
什么时候通知，我着急。他说1周，我说我有offer等，他说尽快。他好像有事，着急就  
走了。

GLM.  这个包括了整个过程，从一开始的data cleaning and transformation, outlier detection, missing value   
related..., etc. 到mulitcollinearity detection, variable/model selection, model fitting 直至最后的model   
validation & model diagnostics...像multicollinearity, model selection, cross-validation这些问题，一定会问，  
屡试不爽。   
  
 2. data mining。需要指出的是，由于regulation的限制，GLM是credit risk modeling的主流方法，fancy的model  
一般来说没啥用武之地(实际上也有论文指出，SVM和boosting之类的方法，对于credit scoring data来说表现并不比  
Logistic Rregression更好)。但也有银行要求有data mining的技术，一般用于marketing和behaviral modeling。一  
般来说，需要的data mining methods会在job requirement里说清楚。这个没啥好说的，太多细节了，自己掌握吧。   
  
 3. SAS. 主要是data step, proc sql, macro. 以及上面说的1和2相对应的proc。   
  
 4. 简历相关的一切技术问题。像我在招工上篇说的，这个答不好是诚信问题。要把简历上的东西和衍生的所有细节都倒  
背如流。   
  
 5. 统计的基本概念，以及如何communicate with non-technical audiences. 比如说，要如何不用任何统计术语来解  
释p-value, confidence interval, model inferences等等。   
  
 6. 其他一些零散的统计和数学知识，比如说experimental design, six sigma, econometrics, optimization..

1. 回答要有层次。举个例子, how to do model selection? 比较一下这三个答案   
     
   A: I use forward/backward/stepwise... blah blah....    
   B: I know three methods, subset selections(forward/backward/stepwise), PCA, shrinkage method   
   (ridge/LASSO). I prefer PCA....blah blah....    
   C: Many people used to prefer forward/backward/stepwise because of its easy computation and   
   straightforward interpretation. However it has some severe problems, e.g. unstable estimations and unable   
   to deal with multicollinearity. Harrell's 2001 paper has detailed discussion. I personally prefer PCA, if there   
   is a good interpretation for those principle components. If it's not the case, shrinkage method may be a   
   better choice. Ridge regression offers a biased but less-variance prediction, however it is not really about   
   "selection" since its shrinking process is continuous. Instead, LASSO truncates some coefficients at 0 and   
   thus discards those correspondent variables, blah...blah....I usually implement those methods in Proc   
   glmselect, blah... blah...  However, the most important are the stories behind the data. Instead of using   
   some fancy statistical stuff, some experts knowledge and business context are more necessary for selecting   
   the right model. blah, blah.......  
     
     interviewer喜欢那个版本的答案应该很清楚了，组织一个既有深度和广度，又有条理的答案会有很大加分。当然这个只  
   是一个例子，我的水平还不够写出足够有深度的答案，希望大牛们可以就这个问题发表自己看法。   
      
   2. 一定要准备好问interviewer的问题。   
    可以反问一些技术问题，比如说，你们在实际中是怎么搞model selection的(有个HM有点尴尬地说他们就在用  
   stepwise, 在我说了stepwise的一堆缺点之后)， 你们怎么evaluate models, etc...   在这里需要严重指出的是，咱们中  
   国人比老美或者阿三强的就是技术，所以在面试里要最大程度地表现出自己的技术优势。但是，不是每个interviewer都  
   有备而来，他们可能只准备了一些很简单的TQ，让你的技术优势无从体现。这时候可以反问一些他们没有问的技术问  
   题，然后在相互讨论中将准备好的答案说出来，这样就可以让interviwers知道你其实 也懂这方面的知识。   
     
    另一类问题是跟职位相关的，比如说这个职位的面临的最大挑战是啥，这个部门在整个公司起到的作用等等。问这类问  
   题，是为了表现1. 你有备而来；2. 你对这个职位很感兴趣。   
     
    还有一些问题需要即席发挥，就interviewer的introduction的内容发问。这个没法准备，不过有个小窍门可以分享－－  
   很多时候，几个interviewer会有内容雷同的introduction。如果你反应不够快不能及时发掘出有意义的问题，可以留到  
   下一个interviewer再问。   
     
    总的来说，问问题的重要性不比答问题的低。而且还有一个窍门在里面--你问的问题多了，interviewer问你的时间自然  
   就少了, then we can take the control and hide our language weakness... 当然要谨记的是，NO STUPID   
   QUESTIONS !!   
     
     3. 抓住一切可以主动表现自己的机会，比如说self introduction和presentation。self intro就不细说了，任何一本面  
   试的书都会介绍应该怎么组织。我觉得presentation需要两点，一要有趣味性，二是要全面。我当时onsite做的ppt尽  
   量减少文字，多用图表，还准备了一个切题的暖场小笑话。至于内容，我一开始先介绍了我现在做的project的理论背景   
   和literature review，尽量简短，没有任何数学推导，只给出结论和reference。然后讨论了simulation和data   
   analysis, 还详细说了一下我在sas中是怎么实现这个比较新的算法的。最后我还说了一下这个算法在credit risk   
   modeling的应用和它的局限性。老板和同事们反应非常热烈，屡屡中途打断我问技术问题，原定半个小时超了半小时。
2. Just had interview with Goldman Sachs,  
   : "Trading Analystics". I think I failed, but I share  
   : with you their questions.  
   :   
   : Stochastic Calculus  
   : 1) Is it obvious that Brownian Motion B(t) is a martingale ?  
   :    and B(t) square ?  
     
   B(t) is, B(t) square is not. The expection of B(t)^2 is t,   
   so B(t)^2 -t is a martingale.  
     
   : 2) Can you integrate B(t) square ? Or which formula do you  
   : use ?  
     
   Use ITO formula.  
     
   :   
   :   
   : Probability  
   :   Two people wants to flip a coin to decide who will use  
   : black  
   :    in a "go" game. They suddenly found that this is not a  
   : fair coin,  
   :    what should they do ?  
     
   Now head-tail and tail-head become equal probability events.  
     
   :   
   : Algorithm  
   : 1) I have a file with unknown length, with characters in it.  
   : You can  
   :    only read the file sequentially. How do you pick a  
   : character, so that  
   :    the chance of picking it, is equal to pick any other one  
   : ?  
     
   The idea is, at any step, you save one character from the  
   previous  
   character set.  
     
   : 2) Give me an example of n\*log(n) sorting algorithm.  
     
   See any C Book.  
     
   :   
   : Programming  
   : 1) What is Virtual Funtion in C++ ?  
     
   See any C++ Book.  
     
   : 2) What is "Associate Arrary" in Perl ?  
     
   See any Perl Book.  
     
   : 3) How do you choose all variable in a table using SQL ?  
     
   create table A as select \* from table B  
     
   :   
   : Computer Basics  
   : 1) What is the Unix command which give you number of lines  
   : in a file ?  
     
   nl  
     
   : 2) What is the Unix command which shows file names in a  
   : directory ?

inheritance vs composition  
2. what is clonable  
3. what is shallow copy & deep copy. how to do deep copy without immplement   
clonable  
4. what is transient  
5. say something about java collection framework. what are the common method  
of  class Collection.  
6. how to immplement a hashmap. how does hashmap immplemented in java  
7. public static final Map m = new HashMap(); can you do m.put("k", "v")?  
8. how does factory pattern work.  
9. some of my previous project  
  
person # 2:  
1. how does java permanate generation & young generation work? (我一下就死菜  
了)  
2. what's the difference of synchronized method and synchronized block。   
what is read/write lock. how does reentrance lock work.  
3. how does ExecutorService work. How does executor recieve/handle the   
status/feedback of each thread.  
4. how does Future work  
5. how does Semaphore & Latch work  
3-5 need to write sample code  
  
person # 3:  
1. how to find intersections of two collections. how to improve the   
performance. what if there are dulicated elements in those collections.  
2. what is trie map, how to build it  
3. some of my previous project  
  
Another group:  
test:  
1. find fibonacci f(n) using efficient algorithm. I used the matrix   
multiplication method but didn't write the correct code. lol  
2. three register, A,B,R, three operations A->R, B->R, (A-R)->R, how to do B  
->A  
3. a very long java code to test your understanding about java argument   
passing  
  
person # 1:  
1. SQL query. left/right inner join  
2. how do you handle stress  
3. how do you handle mistakes during work  
4. lots of other behavior questions  
5. some school project  
  
person # 2:  
1. asked a lot about matrix multiplication for the fibonacci algorithm.   
however I didn't write it correctly so I had a hard time.  
2. java reflection. write a method to return a thread by taking a string of   
class name which immplements runnable interface. what you need to take   
caution when do java reflection. what execptions does java reflection throws  
( I said I don't remember... lol)  
3. Thread.start() Runnable.run() what's the difference  
  
person #3:  
1. explain string literal. will gc recycle a string while it has a reference  
in string literal?  
2. compound SQL queries  
3. HashSet problem  
  
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

SAS Programer Position   
1. What kind of AE tables are there?   
2. What difference between proc means and freq?   
3. What does run statement mean?   
4. What is ITT? What assessment in ITT definition is?   
5. Which procedure can produce standard deviation of a variable?   
6. What do put and input functions do?   
7. How to validate your program?   
8. How to identify TEAE? Namely, what variables can tell us if a patient is   
TEAE?   
9. What differences between scan and substr?   
11. What is MedDRA?   
12. What does the statement break do?   
13. How to code options for estimate and contrast? Are there any differences  
between them?   
14. What PROCs do you use for drawing a graph? How to code for X direct and   
Y direct when   
coding for graphs?   
15. How to remove format?   
16. Is there any difference between macro BBB (a, b) and BBB (a=, b=)?   
17. What Macro functions have you used?   
18. In proc report, what difference between group and order?   
19. How to validate data?   
20. What procedures can perform linear regression analysis?   
21. What procedures can conduct statistical analysis for categorical data?   
22. What difference between Chi-square test and Fisher’s exact test?   
23. Which dataset is on the left hand side in the merging the two datasets A  
and B?   
24. What do we need to pay a special attention to a matching merge?   
25. Have you used ODS? How to organize SAS output (e.g. p-value, odds ratio,  
95% confidence interval) into a table using ODS?   
26. What is %eval for?   
27. How to create a macro?   
28. How to create a macro variable?   
29. How to pass a macro variable?   
30. What difference between global and local macro variable?   
31. What lab tables did you worked on?   
32. Have you accomplished a shift table?   
33. How to avoid changing the raw dataset when you work on proc sort?   
34. Except proc report what else do you use to produce a table?   
35. What is ID statement in Proc transpose?   
36. In proc format, what are value and picture?   
37. What is difference between having and where in proc sql?   
38. What are the two types of parameters that is available in macro? Any   
difference between them?   
39. What kinds of lab datasets are there?   
40. Are there any other datasets have we encountered?   
41. What is macro routine?   
42. When do you use ARRAY, MACRO and SQL?   
43. What macro functions have you used?   
44. How to do validation?   
45. How to implement LOCF?   
46. Why use SAS arrays?   
47.What kind of p-value have you encountered?   
48. What difference between data step functions and macro functions are?   
49. What difference between macro (A=,b=) and macro (A,B)?   
50. Macro debug options   
51. How to code LOCF?   
52. Explain why “NOTE: MERGE statement has more than one data set with   
repeats of BY values.” appears in log file when we merge datasets.   
53. Tell us about your self and your current work.   
54. What is your strength as a SAS programmer?   
55. What is your weak point as a SAS programmer?   
56. How to tell if a program is good or not?   
57. What part of SAS do you like most?   
58. Which part of SAS is more difficult for you?   
59. What is the difference between where and if?   
60. What proc have you used most often?   
61. Data both;   
merge demog (keep=sex race ) vitals;   
run;   
what do you think about this program?   
62. If I have a very long string, how can you get only last character?   
63. What is the most difficult problem when you program and how you solved   
it?   
64. What is difference between sum(a, b) and c=a+b?   
65. What components does the Macro Language contain?   
66. Any difference between %let and call symput ?   
67. What advantages does Macro have?   
68. What options do SAS have? What are differences between statements and   
options of drop, keep, rename?

A SAS technical interview typically starts with a few of the key concepts   
that are essential in SAS programming. These questions are intended to   
separate those who have actual substantive experience with SAS from those   
who have used in only a very limited or superficial way. If you have spent   
more than a hundred hours reading and writing SAS programs, it is safe to   
assume that you are familiar with topics such as these:   
•    SORT procedure   
•    Data step logic   
•    KEEP=, DROP= dataset options   
•    Missing values   
•    Reset to missing, or the RETAIN statement   
•    Log   
•    Data types   
•    FORMAT procedure for creating value formats   
•    IN= dataset option   
Tricky Stuff  
After the interviewer is satisfied that you have used SAS to do a variety of  
things, you are likely to get some more substantial questions about SAS   
processing. These questions typically focus on some of the trickier aspects   
of the way SAS works, not because the interviewer is trying to trick you,   
but to give you a chance to demonstrate your knowledge of the details of SAS  
processing. At the same time, you can show how you approach technical   
questions and issues, and that is ultimately more important than your   
knowledge of any specific feature in SAS.  
STOP statement  
The processing of the STOP statement itself is ludicrously simple. However,   
when you explain the how and why of a STOP statement, you show that you   
understand:  
•    How a SAS program is divided into steps, and the difference   
between a data step and a proc step   
•    The automatic loop in the data step   
•    Conditions that cause the automatic loop to terminate, or to fail  
to terminate   
RUN statement placement  
The output of a program may be different based on whether a RUN statement   
comes before or after a global statement such as an OPTIONS or TITLE   
statement. If you are aware of this issue, it shows that you have written   
SAS programs that have more than the simplest of objectives. At the same   
time, your comments on this subject can also show that you know:  
•    The distinction between data step statements, proc step   
statements, and global statements   
•    How SAS finds step boundaries   
•    The importance of programming style   
SUM or +  
Adding numbers with the SUM function provides the same result that you get   
with the + numeric operator. For example, SUM(8, 4, 3) provides the same   
result as 8 + 4 + 3. Sometimes, though, you prefer to use the SUM function,   
and at other times, the + operator. As you explain this distinction, you can  
show that you understand:  
•    Missing values   
•    Propagation of missing values   
•    Treatment of missing values in statistical calculations in SAS   
•    Why it matters to handle missing values correctly in analytic   
processing   
•    The use of 0 as an argument in the SUM function to ensure that   
the result is not a missing value   
•    The performance differences between functions and operators   
•    Essential ideas of data cleaning   
Statistics: functions vs. proc steps  
Computing a statistic with a function, such as the MEAN function, is not   
exactly the same as computing the same statistic with a procedure, such as   
the UNIVARIATE procedure. As you explain this distinction, you show that you  
understand:  
•    The difference between summarizing across variables and   
summarizing across observations   
•    The statistical concept of degrees of freedom as it relates to   
the difference between sample statistics and population statistics, and the   
way this is implemented in some SAS procedures with the VARDEF= option   
REPLACE= option  
Many SAS programmers never have occasion to use the REPLACE= dataset option   
or system option, but if you are familiar with it, then you have to be aware  
of:  
•    The distinction between the input dataset and the output dataset   
in a step that makes changes in a set of data   
•    The general concept of name conflicts in programming theory   
•    Issues of programming style related to name conflicts   
•    How the system option compares to the corresponding dataset   
option   
A question on this topic may also give you the opportunity to mention syntax  
check mode and issues of debugging SAS programs.  
WHERE vs. IF  
Sometimes, it makes no difference whether you use a WHERE statement or a   
subsetting IF statement. Sometimes it makes a big difference. In explaining   
this distinction, you have the opportunity to discuss:   
•    The distinction between data steps and proc steps   
•    The difference between declaration (declarative) statements and   
executable (action) statements   
•    The significance of the sequence of executable statements in a   
data step   
•    Some of the finer points of merging SAS datasets   
•    A few points of efficiency theory (although tests do not seem to   
bear the theory out in this case)   
•    The origin of the WHERE clause in SQL (of course, bring this up   
only if you’re good at SQL)   
•    WHERE operators that are not available in the IF statement or   
other data step statements   
Compression  
Compressing a SAS dataset is easy to to, so questions about it have more to   
do with determining when it is a good idea. You can weigh efficient use of   
storage space against efficient use of processing power, for example.   
Explain how you use representative data and performance measurements from   
SAS to test efficiency techniques, and you establish yourself as a SAS   
programmer who is ready to deal with large volumes of data. If you can   
explain why compression is effective in SAS datasets and observations larger  
than a certain minimum size and why binary compression works better than   
character compression for some kinds of data, then it shows you take   
software engineering seriously.  
Macro processing  
Almost the only reason interviewers ask about macros is to determine whether  
you appreciate the distinction between preprocessing and processing. Most   
SAS programmers are somewhat fuzzy about this, so if you have it perfectly   
clear in your mind, that makes you a cut about the rest — and if not, at   
least you should know that this is a topic you have to be careful about.   
There are endless technical issues with SAS macros, such as the system   
options that determine how much shows up in the log; your experience with   
this is especially important if the job involves maintaining SAS code   
written with macros.  
SAS macro language is somewhat controversial, so be careful what you say of   
your opinion of it. To some managers, macro use is what distinguishes real   
SAS programmers from the pretenders, but to others, relying on macros all   
the time is a sure sign of a lazy, fuzzy-headed programmer. If you are   
pressed on this, it is probably safe to say that you are happy to work with   
macros or without them, depending on what the situation calls for.  
Procedure vs. macro  
The question, "What is the difference between a procedure and a macro?" can   
catch you off guard if it has never occurred to you to think of them as   
having anything in common. It can mystify you in a completely different way   
if you have thought of procedures and macros as interchangeable parts. You   
might mention:  
•    The difference between generating SAS code, as a macro usually   
does, and taking action directly on SAS data, as a procedure usually does   
•    What it means, in terms of efficiency, for a procedure to be a   
compiled program   
•    The drastic differences in syntax between a proc step and a macro  
call   
•    The IMPORT and EXPORT procedures, which with some options   
generate SAS statements much like a macro   
•    The %SYSFUNC macro function and %SYSCALL macro statement that   
allow a macro to take action directly on SAS data, much like a procedure   
Scope of macro variables  
If the interviewer asks a question about the scope of macro variables or the  
significance of the difference between local and global macro variables,   
the programming concept of scope is being used to see how you handle the new  
ways of thinking that programming requires. The possibility that the same   
name could be used for different things at different times is one of the   
more basic philosophical conundrums in computer programming. If you can   
appreciate the difference between a name and the object that the name refers  
to, then you can probably handle all the other philosophical challenges of   
programming.  
Run groups  
Run-group procedures are not a big part of base SAS, so a question about run  
-group processing and the difference between the RUN and QUIT statements   
probably has more to do with:  
•    What a procedure is   
•    What a step is   
•    All the work SAS has to go through as it alternately acquires a   
part of the SAS program from the execution queue, then executes that part of  
the program   
•    Connecting the program and the log messages   
SAS date values  
Questions about SAS date values have less to do with whether you have   
memorized the reference point of January 1, 1960, than with whether you   
understand the implications of time data treated as numeric values, such as:  
•    Using a date format to display the date variable in a meaningful   
way   
•    Computing a length of time by subtracting SAS date values   
Efficiency techniques  
With today's bigger, faster computers, efficiency is a major concern only   
for the very largest SAS projects. If you get a series of technical   
questions about efficiency, it could mean one of the following:  
•    The employer is undertaking a project with an especially large   
volume of data   
•    The designated computer is not one of today's bigger, faster   
computers   
•    The project is weighed down with horrendously inefficient code,   
and they are hoping you will be able to clean it all up   
On the other hand, the interviewer may just be trying to gauge how well you   
understand the way SAS statements correspond to the actions the computer   
takes or how seriously you take the testing process for a program you write.  
Debugger  
Most SAS programmers never use the data step debugger, so questions about it  
are probably intended to determine how you feel about debugging — does the  
debugging process bug you, or is debugging one of the most essential things  
you do as a programmer?  
Informats vs. formats  
If you appreciate the distinction between informats and formats, it shows   
that:  
•    You can focus on details   
•    It doesn't confuse you that two routines have the same name   
•    You have some idea of what is going on when a SAS program runs   
TRANSPOSE procedure  
The TRANSPOSE procedure has a few important uses, but questions about it   
usually don't have that much to do with the procedure itself. The intriguing  
characteristic of the TRANSPOSE procedure is that input data values   
determine the names of output variables. The implication of this is that if   
the data values are incorrect, the program could end up with the wrong   
output variables. In what other ways does a program depend on having valid   
or correct data values as a starting point? What does it take to write a   
program that will run no matter what input data values are supplied?  
\_N\_  
Questions about the automatic variable \_N\_ (this might be pronounced “  
underscore N underscore” or just “N”) are meant to get at your   
understanding of the automatic actions of the data step, especially the   
automatic data step loop, also known as the observation loop.  
A possible follow-up question asks how you can store the value of \_N\_ in the  
output SAS dataset. If you can answer this, it may show that you know the   
properties of automatic variables and know how to create a variable in the   
data step.  
PUT function  
A question about the PUT function might seem to be a trick question, but it   
is not meant to be. Beyond showing that you aren't confused by two things as  
different as a statement and a function having the same name, your   
discussion of the PUT function can show:   
•    An understanding of what formats are   
•    Your experience in creating variables in data step statements   
Important SAS trivia  
Some SAS trivia may be important to know in a technical interview, even   
though it may never come up in your actual SAS programming work.   
•    MERGE is a data step statement only. There is no MERGE procedure.  
“PROC MERGE” is a mythical construction created years ago by Rhena   
Seidman, and if you are asked about it in a job interview, it is meant as a   
trick question.   
•    It is possible to use the MERGE statement without a BY statement,  
but this usually occurs by mistake.   
•    SAS does not provide an easy way to create a procedure in a SAS   
program. However, it is easy to define informats and formats and use them in  
the same program. Beginning with SAS 9.2, the same is true of functions.   
•    The MEANS and SUMMARY procedures are identical except for the   
defaults for the PRINT option and VAR statement.   
•    Much of the syntax of the TABULATE procedure is essentially the   
same of that of the SUMMARY procedure.   
•    CARDS is another name for DATALINES (or vice versa).   
•    “DATA \_NULL\_” is commonly used as a code word to refer to data   
step programming that creates print output or text data files.   
•    The program data vector (PDV) is a logical block of data that   
contains the variables used in a data step or proc step. Variables are added  
to the program data vector in order of appearance, and this is what   
determines their position (or variable number) attribute.