关于本面经与代码合集

http://www.mitbbs.com/article_t1/JobHunting/31820505_0_1.html

原文为 evaeva 网友发表于 mitbbs, 原文一共有 7 个 doc 文档，应该是对 mitbbs jobhutting 版截至 2011-03 之前面经的大合集，这里转化成 pdf，主要是方便查询。

http://www.mitbbs.com/article_t/JobHunting/31505215.html

还有一个部分是 mitbbs59 网友整理于 2010-01 的大合集。

本人的一些经验：

1. http://csapp.cs.cmu.edu/ <Computer Systems: A Programmer's Perspective> 是一本非常好的 computer science 入门的教科书，网上可以找到第一版的 djvu 版本。
2. 本人强烈否定推荐 http://mitpress.mit.edu/algorithms/ <Introduction to Algorithms>, 网上可以找到第二版的 chm 版本，原因很简单，对于面试准备，不实用。这是一本非常好的 Algorithms 经典教程，mit ocw 也有大牛上课时的视频。
3. http://www.algolist.net/ & http://cslibrary.stanford.edu/ 简单明了的 tutorial 教程; http://www.cs.bell-labs.com/cm/cs/pearls/ 老一代的程序员基本都读过这本书，网上只有第一版的 djvu 版本.
4. Algorithm Design, 对于象 Google, Facebook 这样的公司，会涉及很多这样的问题，网上可以找到的教科书，主要是 <Algorithm Design> Cornell University 2006 版本， <Introduction to the design and analysis of algorithms> Levitin 中文翻译第二版, <Algorithms> Berkeley 2006 版本，这个有 free 的 copy. http://www.cs.berkeley.edu/~vazirani/algorithms.html
5. OOD，Amazon 这样的公司非常喜欢问这类问题，可能对面试准备有用的书是，<design pattern> <code complete> etc …
6. 最后就是面经了，mitbbs jobhutting, <Programming Interview Exposed> <Careercup 150 问> <ihas1337code.com> <geeksforgeeks.org> <sureinterview.com> <编程之美> <程序员面试宝典> <何海涛面试题精选> etc

Heap sort:

```
#include "stdafx.h"
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

void HeapSort(int num[],int size);

 void BuildHeap(int num[] ,int size);

void PercolateDown(int num[] , int index,int size);

 void PrintHeap(const char* strMsg,int array[],int nLength);

 void Swap(int num[] , int v, int u);

     int main(int argc, char *argv[])

{

 int data[13]={8,5,4,6,13,7,1,9,12,11,3,10,2};

 HeapSort(data,13);

  system("PAUSE");

  return 0;

 }


void HeapSort(int num[] ,int size)

 {

    int i;

    int iLength=size;

     PrintHeap("Befor Sort:",num,iLength);

     BuildHeap(num,size);// 建立小顶堆

   for (i = iLength - 1; i >= 1; i--) {

       Swap(num, 0, i);// 交换

        size--;// 每交换一次让规模减少一次

        PercolateDown(num, 0,size);// 将新的首元素下滤操作

       PrintHeap("Sort Heap:",num,iLength);

   }
 }
  // 建堆方法，只需线性时间建好
void BuildHeap(int num[] ,int size) {
     int i;
     for (i = size / 2 - 1; i >= 0; i--) {//
对前一半的节点（解释为“从最后一个非叶子节点开始，将每个父节点都调整为最小堆”更合理一些）
         PercolateDown(num, i,size);// 进行下滤操作
       PrintHeap("Build heap:",num,size);
    }
 }


 // 对该数进行下滤操作，直到该数比左右节点都小就停止下滤
 void PercolateDown(int num[] , int index,int size) {
     int min;// 设置最小指向下标
     while (index * 2 + 1<size) {// 如果该数有左节点，则假设左节点最小
      min = index * 2 + 1;// 获取左节点的下标
       if (index * 2 + 2<size) {// 如果该数还有右节点
            if (num[min] > num[index * 2 + 2]) {// 就和左节点分出最小者
                min = index * 2 + 2;// 此时右节点更小，则更新min的指向下标
            }
        }

        if (num[index] < num[min]) {// 如果index最小，
            break;// 停止下滤操作
        } else {
            Swap(num, index, min);// 交换两个数，让大数往下沉
             index = min;// 更新index的指向
         }
    }// while
 }
```

```
// 给定数组交换两个数的位置

 void Swap(int num[] , int v, int u) {

     int temp = num[v];

    num[v] = num[u];

     num[u] = temp;

 }


void PrintHeap(const char* strMsg,int array[],int nLength)

{

      int i;

    printf("%s",strMsg);

      for(i=0;i<nLength;i++)

      {

        printf("%d ",array[i]);

     }

     printf("\n");

 }
```

Quicksort:

```
#include "stdafx.h"


#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;
void Swap(int num[] , int v, int u);
void qSort(int num[],int left, int right);

int main(int argc, char *argv[])

{

 int data[13]={8,5,4,6,13,7,1,9,12,11,3,10,2};
 for(int i=0;i<13;i++)
               cout<<data[i]<<"---";

   qSort(data,0,12);

   for(int i=0;i<13;i++)
               cout<<data[i]<<"---";
 return 0;

 }

void qSort(int num[],int l, int r)
{   int pivot=(l+r)/2;

   int left=l; int right=r;
   Swap(num,pivot,left);

        while( left<=right)
        {  while(num[left]<=num[l]) left++;
          while(num[right]>=num[l]) right--;
          if(left<right)
          { Swap(num,left,right);
          left++;right--;}
         }

   Swap(num,--left,l);

        if(l<left-1)
        qSort(num,l,left-1);
        if(left+1<r)
        qSort(num,left+1,r);
}
```

Bubble Sort

```
#include "stdafx.h"


#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;
void Swap(int num[] , int v, int u);
void qSort(int num[],int l);

int main(int argc, char *argv[])

{

 int data[13]={8,5,4,6,13,7,1,9,12,11,3,10,2};
int l=sizeof(data)/sizeof(int);

 for(int i=0;i<13;i++)
          cout<<data[i]<<"---";
  cout<<endl;

  qSort(data,l);

  for(int i=0;i<13;i++)
          cout<<data[i]<<"---";



 return 0;

 }
```

```
void qSort(int* num,int l)
{

        for (int i=0;i<l;i++)
                for( int j=i;j<l;j++)
                        if (num[i]>num[j])
             Swap(num,i,j);
}



void Swap(int num[] , int v, int u)
{ int temp;
  temp=num[v];;
  num[v]=num[u];
  num[u]=temp;
}
```

BST:

```
#include "stdafx.h"
#include <iostream>
#include <list>
#include <algorithm>
//#include <cstdlib>
using namespace std;





class BinarySearchTree
{
   private:
      struct tree_node
      {
        tree_node* left;
        tree_node* right;
        int data;
      };
      tree_node* root;

   public:
      BinarySearchTree()
      {
        root = NULL;
      }

      bool isEmpty() const { return root==NULL; }
      void print_inorder();
      void inorder(tree_node*);
      void print_preorder();
      void preorder(tree_node*);
      void print_postorder();
      void postorder(tree_node*);
      void insert(int);
      void remove(int);
};
```

```
// Smaller elements go left
// larger elements go right
void BinarySearchTree::insert(int d)
{
   tree_node* t = new tree_node;
   tree_node* parent;
   t->data = d;
   t->left = NULL;
   t->right = NULL;
   parent = NULL;

   // is this a new tree?
   if(isEmpty()) root = t;
   else
   {
      //Note: ALL insertions are as leaf nodes
      tree_node* curr;
      curr = root;
      // Find the Node's parent
      while(curr)
      {
         parent = curr;
         if(t->data > curr->data) curr = curr->right;
         else curr = curr->left;
      }

      if(t->data < parent->data)
        parent->left = t;
      else
        parent->right = t;
   }
}

void BinarySearchTree::remove(int d)
{
   //Locate the element
   bool found = false;
   if(isEmpty())
   {
      cout<<" This Tree is empty! "<<endl;
      return;
   }

   tree_node* curr;
   tree_node* parent;
   curr = root;

   while(curr != NULL)
   {
       if(curr->data == d)
       {
         found = true;
         break;
       }
       else
       {
          parent = curr;
          if(d>curr->data) curr = curr->right;
          else curr = curr->left;
       }
   }
   if(!found)
                 {
      cout<<" Data not found! "<<endl;
      return;
   }


   if((curr->left == NULL && curr->right != NULL)|| (curr->left != NULL
&& curr->right == NULL))
   {
     if(curr->left == NULL && curr->right != NULL)
     {
        if(parent->left == curr)
        {
          parent->left = curr->right;
          delete curr;
        }
        else
        {
          parent->right = curr->right;
          delete curr;
        }
     }
     else // left child present, no right child
     {
       if(parent->left == curr)
        {
          parent->left = curr->left;
          delete curr;
        }
        else
        {
          parent->right = curr->left;
          delete curr;
        }
     }
    return;
   }

                 //We're looking at a leaf node
                 if( curr->left == NULL && curr->right == NULL)
   {
      if(parent->left == curr) parent->left = NULL;
      else parent->right = NULL;
                                 delete curr;
                                 return;
   }


   //Node with 2 children
   // replace node with smallest value in right subtree
   if (curr->left != NULL && curr->right != NULL)
   {
      tree_node* chkr;
      chkr = curr->right;
      if((chkr->left == NULL) && (chkr->right == NULL))
      {
            curr = chkr;
            delete chkr;
            curr->right = NULL;
        }
        else // right child has children
        {
            //if the node's right child has a left child
            // Move all the way down left to locate smallest element

            if((curr->right)->left != NULL)
            {
                tree_node* lcurr;
                tree_node* lcurrp;
                lcurrp = curr->right;
                lcurr = (curr->right)->left;
                while(lcurr->left != NULL)
                {
                   lcurrp = lcurr;
                   lcurr = lcurr->left;
                }
                       curr->data = lcurr->data;
                delete lcurr;
                lcurrp->left = NULL;
           }
           else
           {
               tree_node* tmp;
               tmp = curr->right;
               curr->data = tmp->data;
                  curr->right = tmp->right;
               delete tmp;
           }

        }
                        return;
    }

}

void BinarySearchTree::print_inorder()
{
  inorder(root);
}

void BinarySearchTree::inorder(tree_node* p)
{
    if(p != NULL)
    {
        if(p->left) inorder(p->left);
        cout<<" "<<p->data<<" ";
        if(p->right) inorder(p->right);
    }
    else return;
}

void BinarySearchTree::print_preorder()
{
    preorder(root);
}
```

```
void BinarySearchTree::preorder(tree_node* p)
{
   if(p != NULL)
   {
      cout<<" "<<p->data<<" ";
      if(p->left) preorder(p->left);
      if(p->right) preorder(p->right);
   }
   else return;
}

void BinarySearchTree::print_postorder()
{
   postorder(root);
}

void BinarySearchTree::postorder(tree_node* p)
{
   if(p != NULL)
   {
      if(p->left) postorder(p->left);
      if(p->right) postorder(p->right);
      cout<<" "<<p->data<<" ";
   }
   else return;
}

int main()
{
   BinarySearchTree b;
   int ch,tmp,tmp1;
   while(1)
   {
     cout<<endl<<endl;
     cout<<" Binary Search Tree Operations "<<endl;
     cout<<" ----------------------------- "<<endl;
     cout<<" 1. Insertion/Creation "<<endl;
     cout<<" 2. In-Order Traversal "<<endl;
     cout<<" 3. Pre-Order Traversal "<<endl;
     cout<<" 4. Post-Order Traversal "<<endl;
     cout<<" 5. Removal "<<endl;
     cout<<" 6. Exit "<<endl;
     cout<<" Enter your choice : ";
     cin>>ch;
     switch(ch)
     {
        case 1 : cout<<" Enter Number to be inserted : ";
                 cin>>tmp;
                 b.insert(tmp);
                 break;
        case 2 : cout<<endl;
                 cout<<" In-Order Traversal "<<endl;
                 cout<<" -------------------"<<endl;
                 b.print_inorder();
                 break;
        case 3 : cout<<endl;
                 cout<<" Pre-Order Traversal "<<endl;
                 cout<<" -------------------"<<endl;
```

```
               b.print_preorder();
               break;
        case 4 : cout<<endl;
               cout<<" Post-Order Traversal "<<endl;
               cout<<" --------------------"<<endl;
               b.print_postorder();
               break;
        case 5 : cout<<" Enter data to be deleted : ";
               cin>>tmp1;
               b.remove(tmp1);
               break;
        case 6 :
               return 0;

     }
   }
}
```

Print all subsets:

```
#include "stdafx.h"


#include <vector>
using namespace std;

void PrintSubsetsInt(const vector<int> &set,
   int pos,
   int len,
   int start,
   vector<int> &output)
{
   if (len == 0)
   {
      for (int i = 0; i < output.size(); ++i)
      {
         printf("%d ", output[i]);
      }
      printf("\r\n");
      return;
   }

   for (int i = start; i < set.size(); ++i)
   {
      output[pos] = set[i];
      PrintSubsetsInt(set, pos + 1, len - 1, i + 1, output);
   }
}

void PrintAllSubsets(const vector<int> &set)
{
   for (int i = 0; i < set.size(); ++i)
   {
      vector<int> output(i);
      PrintSubsetsInt(set, 0, i, 0, output);
   }
}

int _tmain(int argc, _TCHAR* argv[])
{
   vector<int> x(8);
   for (int i = 0; i < x.size(); i++)
   {
      x[i] = i;
   }

   PrintAllSubsets(x);

   return 0;
}
//////////////////////////////////////////////////////////////////////////////////////
#include "stdafx.h"

#include <iostream>

using namespace std;

void Swap(char n[], int begin, int end)
{   char temp;
        while( begin<end)
        {
          temp=n[begin];
          n[begin]=n[end];
          n[end]=temp;
         begin++;
         end--;
        }
}


void Reverse(char s[])
{
        int start=0;
        int end=0;
        int length=strlen(s);
        Swap(s,start,length-1);

        while(end<length)
        { if (s[end]!=' ')
          {
                start=end;
                while( end<length && s[end]!=' ')
                        end++;
                Swap(s,start,--end);
        }
        end++;
        }
}


void main()
{
        char s[]="C# strings are almost identical to Java strings";


    Reverse(s);
        cout<<s<<endl;
}
//////////////////////////////////////////////////
#include "stdafx.h"
```

```
#include <iostream>

using namespace std;

void atoi(char* s)
{
        int start=0;
        bool isNeg=false;
        int n=0;
        if(s[start]=='-')
        {isNeg=true;
        start++;}
        while(s[start])
  { int m=s[start++]-'0';
    n=10*n+m;
  }
 if(isNeg)
          n=0-n;
 cout<<n<<endl;
}
void main()
{
        char* s="-12345";
        atoi(s);
}
//////////////////////////////////////////Binary seearch
#include "stdafx.h"
#include <string>
#include <iostream>
#include <vector>
using namespace std;

void bs(int* n,int s,int end, int i)
{   if(s>=end)
{ cout<<"not found";
return;}
        int m=(s+end)/2;
   if(n[s+m]<i)
                bs(n,s+m+1,end,i);
        else if (n[s+m]>i)
                bs(n,s,s+m-1,i);
        else
                cout<<"find i:";
}

void main()
{ int n[]={1,2,3,4,5,6,7,8};
 bs(n,0,sizeof(n)/sizeof(int),10);
}
///////////////////////////////////////////////String Permutation:
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;

void swap(char* first, char* second)
{
      char ch = *second;
```

```
        *second = *first;
        *first = ch;
}

int permute(char* set, int begin, int end)
{
        int i;
        int range = end - begin;
        if (range == 1) {
                cout << set << endl;
        } else {
                for(i=0; i<range; i++) {
                        swap(&set[begin], &set[begin+i]);                 //initial swap
                        permute(set, begin+1, end);                               //recursion
                        swap(&set[begin], &set[begin+i]);        //swap back
                }
        }
        return 0;
}

//Example Implementation -- Up to you on how to use
int main()
{
        char str[255]; //string
           cout << "Please enter a string: ";
           cin.getline(str, 255); //take string
        permute(str, 0, strlen(str)); //permutate the string
        return 0;
}
///////////////////////////////////////////////////////////////////Reverse Bit:
unsigned char rev( unsigned char b ) {
  unsigned char result = 0;
  while (b) {
    result <<= 1;

    result  |= b % 2;
    b      >>= 1;
    }
  return result;
  }

void main()
{
           cout<< rev('g');
}
```

Another:

```
int mirror_bits( int num)
{
     int tmp=0;
     int i=0;
     int int_size_in_bits=sizeof(int)*8;

    while(num>(1<<i)){
        if((1<<i)&num){
            tmp |= 1<<(int_size_in_bits-1-i);
        }
        i++;
    }
    return tmp;
}
FAST:
unsigned int reverseBits(unsigned int x) {
  x = (((x & 0xaaaaaaaa) >> 1) | ((x & 0x55555555) << 1));
  x = (((x & 0xcccccccc) >> 2) | ((x & 0x33333333) << 2));
  x = (((x & 0xf0f0f0f0) >> 4) | ((x & 0x0f0f0f0f) << 4));
  x = (((x & 0xff00ff00) >> 8) | ((x & 0x00ff00ff) << 8));
  return((x >> 16) | (x << 16));
}
//////////////////////////////////////////////////////////Rotated Binary:
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;

int rotate(int* n, int s,int e)
{   if(n[s]>=n[e]&&s<e)
  {
        int m=(s+e)/2;
        cout<<m<<endl;
    if( n[m]<n[s])
                return rotate(n,s,m);
        else return rotate(n,m+1,e);
   }
  else return s;
}

void main()
{ int a[14]={4,5,6,7,8,9,10,11,12,13,14,1,2,3};
  cout<<rotate(a,0,13);
}
////////////////////////////
Hashtable setupDictionary(string[] book) {
Hashtable table = new Hashtable();
foreach (string word in book) {
if (!table.contains(word)) {
table.add(word, 0);
}
table[word] = table[word] + 1;
}
}
int getFrequency(Hashtable table, string word) {
if (table == null or word == null) {
return -1;
}
if (table.contains(word)) {
return table[word];
}
return 0;
}
////////////////////////////////////

public static Character firstNonRepeated( String str ){
Hashtable charHash = new Hashtable();
int i, length;
Character c;
Integer intgr;
length = str.length();
// Scan str, building hash table
for (i = 0; i < length; i++) {
c = new Character(str.charAt(i));
intgr = (Integer) charHash.get(c);
if (intgr == null) {
charHash.put(c, new Integer(1));
} else {
// Increment count corresponding to c
charHash.put(c, new Integer(intgr.intValue() + 1));
}
}
// Search hashtable in order of str
for (i = 0; i < length; i++) {
c = new Character(str.charAt(i));
if (((Integer)charHash.get(c)).intValue() == 1)
return c;
}
return null;
string removeChars( string str, string remove ){
char[] s = str.toCharArray();
char[] r = remove.toCharArray();
bool[] flags = new bool[128]; // assumes ASCII!
int len = s.Length;
int src, dst;
// Set flags for characters to be removed
for( src = 0; src < len; ++src ){
flags[r[src]] = true;
}
src = 0;
dst = 0;
// Now loop through all the characters,
// copying only if they aren't flagged
while( src < len ){
if( !flags[ (int)s[src] ] ){
s[dst++] = s[src];
}
++src;
}
return new string( s, 0, dst );
}
void permute( String str ){
int length = str.length();
boolean[] used = new boolean[ length ];
StringBuffer out = new StringBuffer();
char[] in = str.toCharArray();
doPermute( in, out, used, length, 0 );
}
void doPermute( char[] in, StringBuffer out,
boolean[] used, int length, int level ){
if( level == length ){
System.out.println( out.toString() );
return;
}
for( int i = 0; i < length; ++i ){
if( used[i] ) continue;
out.append( in[i] );
used[i] = true;
doPermute( in, out, used, length, level + 1 );
used[i] = false;
out.setLength( out.length() - 1 );
}
}

void combine( string str ){
int length = str.Length;
char[] instr = str.ToCharArray();
StringBuilder outstr = new StringBuilder();
doCombine( instr, outstr, length, 0, 0 );
}
```

```
void doCombine( char[] instr, StringBuilder outstr, int length,
int level, int start ){
for( int i = start; i < length; i++ ){
outstr.Append( instr[i] );
Console.WriteLine( outstr );
if( i < length - 1 ){
doCombine( instr, outstr, length, level + 1, i + 1 );
}
outstr.Length = outstr.Length - 1;
}
}
```

```
static final int PHONE_NUMBER_LENGTH = 7;
void printTelephoneWords( int[] phoneNum ){
char[] result = new char[ PHONE_NUMBER_LENGTH ];
doPrintTelephoneWords( phoneNum, 0, result );}
void doPrintTelephoneWords( int[] phoneNum, int curDigit,
char[] result ){
if( curDigit == PHONE_NUMBER_LENGTH ){
System.out.println( new String( result ) );
return;
}
for( int i = 1; i <= 3; i++ ){
result[ curDigit ] = getCharKey( phoneNum[curDigit], i );
doPrintTelephoneWords( phoneNum, curDigit + 1, result );
if( phoneNum[curDigit] == 0 ||
phoneNum[curDigit] == 1) return;
}
}
```

```
static final int PHONE_NUMBER_LENGTH = 7;
void printTelephoneWords( int phoneNum[] ){
char[] result = new char[ PHONE_NUMBER_LENGTH ];
int i;
/* Initialize the result (in our example,
* put GWP1WAR in result).
*/
for( i = 0; i < PHONE_NUMBER_LENGTH; i++ )
result[i] = getCharKey( phoneNum[i], 1 );
/* Main loop begins */
while( true ){
for( i = 0; i < PHONE_NUMBER_LENGTH; ++i ){
System.out.print( result[i] );
}
System.out.print( ‘\n’ );
/* Start at the end and try to increment from right
* to left.
*/
for( i = PHONE_NUMBER_LENGTH - 1; i >= -1; i-- ){
/* You’re done because you
* tried to carry the leftmost digit.
*/
if( i == -1 ) return;
/* Otherwise, we’re not done and must continue. */
/* You want to start with this condition so that you can
* deal with the special cases, 0 and 1 right away.
*/
if( getCharKey( phoneNum[i], 3 ) == result[i] ||
phoneNum[i] == 0 || phoneNum[i] == 1 ){
result[i] = getCharKey( phoneNum[i], 1 );
/* No break, so loop continues to next digit */
} else if ( getCharKey( phoneNum[i], 1 ) == result[i] ){
result[i] = getCharKey( phoneNum[i], 2 );
break;
} else if ( getCharKey( phoneNum[i], 2 ) == result[i] ){
result[i] = getCharKey( phoneNum[i], 3 );
break;
}
}
}
}}
boolean overlap( Rect a, Rect b){
return( a.ul.x <= b.lr.x &&
a.ul.y >= b.lr.y &&
a.lr.x >= b.ul.x &&
a.lr.y <= b.ul.y );
}
```

To ensure that you didn’t make a mistake, it’s a good idea to verify that these conditions make sense.

The preceding function determines that two rectangles overlap if

❑ A’s left edge is to the left of B’s right edge and
❑ A’s upper edge is above B’s bottom edge and
❑ A’s right edge is to the right of B’s left edge and
❑ A’s bottom edge is below B’s upper edge.

```
bool endianness(){
int testNum;
char *ptr;
testNum = 1;
ptr = (char *) &testNum;
return (*ptr); /* Returns the byte at the lowest address */
}
bool endianness(){
union {
int theInteger;
char singleByte;
} endianTest;
endianTest.theInteger = 1;
return endianTest.singleByte;
}

int numOnesInBinary( int number )
{
int numOnes = 0;
while( number != 0 ){
if( ( number & 1 ) == 1 )
numOnes++;
number = number >>> 1;
}
return numOnes;
}
int numOnesInBinary( int number ){
int numOnes = 0;
while( number != 0 ){
number = number & (number – 1);
numOnes++;
}
return numOnes;
}
```

情形一：如果栈不空并且当前将要入栈的柱子比栈顶的柱子低，则有：

1. 栈顶柱子的右边界完全确定，其对应的局部最大矩形面积可求(下面说明了左边界在它入栈时已确定)。更新全局最大矩形面积后，栈顶柱子可以依次出栈，直到当前的栈顶柱子比当前要入栈的柱子低或者栈变为空栈。

2. 连续的出栈操作使得当前的栈顶柱子比当前要入栈的柱子低或者栈变为空栈, 这时候，当前要入栈的柱子的左边界也确定,可以入栈。

情形二：如果栈为空或者当前要入栈的柱子比栈顶柱子高，则无出栈操作，且当前要入栈的柱子的左边界确定。

总结：

1. 每个入栈操作, 如果入栈柱子低于栈顶柱子,则它确定了栈中比要入栈柱子高的那些柱子的右边界,可以对它们执行出栈操作。出栈的过程伴随着矩形面积的计算。
2. 每个入栈操作，不论是否引起出栈操作,我们都可以确定当前要入栈柱子的左边界。
3. 每次入栈后,栈内所剩柱子一定保持高度单调非递减的顺序。
4. 可令最后一根柱子高度为-1, 保证必有出栈操作。
5. 除了高度为-1的哑元柱子, 所有的柱子都入栈一次，出栈一次，复杂度为O(n)
6. 对高度相同的两根柱子，我们可视后面的柱子比前面的柱子高。

```
struct HistElem
{
   int index; // Index of the element, 0, 1, 2, ...
   int height; // Height of the element
   int far_length_left; // How far can we extend to the left boundary
};

// The last element of the histogram is set to the dummy value -1
// This can be used to indicate that we have reached the right boundary
int HistogramMaxRect(int Hist[], int n)
{
   if (n < 1 || Hist[n-1] != -1)
   {
     printf("Incorrect input.\n");
     return -1;
   }

   int S = 0;
   struct HistElem Stack[MAX_SIZE];
   int top = -1;

   for (int i = 0; i < n; i++)
   {
      // Look backward for elements available for rectangle area calculation
      while (top != -1 && Hist[i] < Stack[top].height)
      {
         int area = Stack[top].height *
                (Stack[top].far_length_left + (i - Stack[top].index));
         if (area > S)
            S = area;

         // We can pop up these elements now
         top--;
      }

      // Find far_length_left for current element
      int far_length_left = i;
      if (top != -1)
         far_length_left = i - Stack[top].index - 1;

      // Push current element into the stack
      top++;
      if (top == MAX_SIZE)
      {
         printf("Stack overflow.\n");
         return -1;
      }
```

```
        Stack[top].index = i;
        Stack[top].height = Hist[i];
        Stack[top].far_length_left = far_length_left;
    }

    return S;
}

int main()
{
    // May change to other test cases
    int Hist[] = {5, 13, 10, 0, 7, 7, 8, -1};
    int S = HistogramMaxRect(Hist, 8);
    printf("The largest size is %d\n", S);
    return 0;
}


struct HistElem
{
    int index;
    int leftWidth;
};
int MaxRect(int * H, int n)
{
   struct HistElem Stack[MAX_SIZE];
   int top=-1;
   int area,maxArea=0;
   for(int pos=0;pos<n;pos++) {
       while(top!=-1 && H[pos]<H[Stack[top].index]{
        area=H[Stack[top].index]*(Stack[top].leftWidth+pos-Stack[top].index);
        if(area>maxArea) maxArea=area;
            top--;
       }
       int leftWidth=0;
       if(top!=-1 && H[pos]>=H[Stack[top].index]) {
            leftWidth=pos-Stack[top].index-1;
       }
       top++;
       Stack[top].leftWidth=leftWidth;
       Stack[top].index=pos;
       }
   }
   while(top!=-1){
      area=H[Stack[top].index]*(Stack[top].leftWidth+n-Stack[top].index);
      if(area>maxArea) maxArea=area;
      top--;
   }
   return maxArea;
}
```

```
void itoa(int n, char s[])
{
int i, sign;
if ((sign = n) < 0) /* record sign */
n = -n; /* make n positive */
i = 0;
do { /* generate digits in reverse order */
s[i++] = n % 10 + '0'; /* get next digit */
} while ((n /= 10) > 0); /* delete it */
if (sign < 0)
s[i++] = '-';
s[i] = '\0';
reverse(s);}

/* trim: remove trailing blanks, tabs, newlines */
int trim(char s[])
{
int n;
for (n = strlen(s)-1; n >= 0; n--)
if (s[n] != ' ' && s[n] != '\t' && s[n] != '\n')
break;
s[n+1] = '\0';
return n;
}
/* strindex: return index of t in s, -1 if none */
int strindex(char s[], char t[])
{
int i, j, k;
for (i = 0; s[i] != '\0'; i++) {
for (j=i, k=0; t[k]!='\0' && s[j]==t[k]; j++, k++);
if (k > 0 && t[k] == '\0')
return i;
}
return -1;
}
#include <ctype.h>
/* atof: convert string s to double */
double atof(char s[])
{
double val, power;
int i, sign;
for (i = 0; isspace(s[i]); i++) /* skip white space */
;
sign = (s[i] == '-') ? -1 : 1;
if (s[i] == '+' || s[i] == '-')
i++;
for (val = 0.0; isdigit(s[i]); i++)
val = 10.0 * val + (s[i] - '0');
if (s[i] == '.')
i++;
for (power = 1.0; isdigit(s[i]); i++) {
val = 10.0 * val + (s[i] - '0');
power *= 10;
}
return sign * val / power;
}
/* qsort: sort v[left]...v[right] into increasing order */
```

```
void qsort(int v[], int left, int right)
{
int i, last;
void swap(int v[], int i, int j);
if (left >= right) /* do nothing if array contains */
return; /* fewer than two elements */
swap(v, left, (left + right)/2); /* move partition elem */
last = left; /* to v[0] */
for (i = left + 1; i <= right; i++) /* partition */
if (v[i] < v[left])
swap(v, ++last, i);
swap(v, left, last); /* restore partition elem */
qsort(v, left, last-1);
qsort(v, last+1, right);
}
/* strcpy: copy t to s; pointer version 2 */
void strcpy(char *s, char *t)
{
while ((*s++ = *t++) != '\0')
;
}
/* strcmp: return <0 if s<t, 0 if s==t, >0 if s>t */
int strcmp(char *s, char *t)
{
for ( ; *s == *t; s++, t++)
if (*s == '\0')
return 0;
return *s - *t;
}

#include <stdio.h>
/* cat: concatenate files, version 1 */
main(int argc, char *argv[])
{
FILE *fp;
void filecopy(FILE *, FILE *)
if (argc == 1) /* no args; copy standard input */
filecopy(stdin, stdout);
else
while(--argc > 0)
if ((fp = fopen(*++argv, "r")) == NULL) {
printf("cat: can't open %s\n, *argv);
return 1;
} else {
filecopy(fp, stdout);
fclose(fp);
}
return 0;
}
/* filecopy: copy file ifp to file ofp */
void filecopy(FILE *ifp, FILE *ofp)
{
int c;
while ((c = getc(ifp)) != EOF)
putc(c, ofp);
}
/* fgets: get at most n chars from iop */
```

```
char *fgets(char *s, int n, FILE *iop)
{
register int c;
register char *cs;
cs = s;
while (--n > 0 && (c = getc(iop)) != EOF)
if ((*cs++ = c) == '\n')
break;
*cs = '\0';
return (c == EOF && cs == s) ? NULL : s;
}

/* fputs: put string s on file iop */
int fputs(char *s, FILE *iop)
{
int c;
while (c = *s++)
putc(c, iop);
return ferror(iop) ? EOF : 0;}
#include "syscalls.h"
main() /* copy input to output */
{
char buf[BUFSIZ];
int n;
while ((n = read(0, buf, BUFSIZ)) > 0)
write(1, buf, n);
return 0;
}
#include <stdio.h>
#include <fcntl.h>
#include "syscalls.h"
#define PERMS 0666 /* RW for owner, group, others */
void error(char *, ...);
/* cp: copy f1 to f2 */
main(int argc, char *argv[])
{
int f1, f2, n;
char buf[BUFSIZ];
if (argc != 3)
error("Usage: cp from to");
if ((f1 = open(argv[1], O_RDONLY, 0)) == -1)
error("cp: can't open %s", argv[1]);
if ((f2 = creat(argv[2], PERMS)) == -1)
error("cp: can't create %s, mode %03o",
argv[2], PERMS);
while ((n = read(f1, buf, BUFSIZ)) > 0)
if (write(f2, buf, n) != n)
error("cp: write error on file %s", argv[2]);
return 0;
}
#include "syscalls.h"
/*get: read n bytes from position pos */
int get(int fd, long pos, char *buf, int n)
{
if (lseek(fd, pos, 0) >= 0) /* get to pos */
return read(fd, buf, n);
else
return -1;
}
template<class T>
class priqueue {
private:
int n, maxsize;
T *x;
void swap(int i, int j)
{ T t = x[i]; x[i] = x[j]; x[j] = t; }
public:
priqueue(int m)
{ maxsize = m;
x = new T[maxsize+1];
n = 0;
}
void insert(T t)
{ int i, p;
x[++n] = t;
for (i = n; i > 1 && x[p=i/2] > x[i]; i = p)
swap(p, i);
}
T extractmin()
{ int i, c;
T t = x[1];
x[1] = x[n--];
for (i = 1; (c = 2*i) <= n; i = c) {
if (c+1 <= n && x[c+1] < x[c])
c++;
if (x[i] <= x[c])
break;
swap(c, i);
}
return t;}};
```

Longest duplicate string:

```
while (ch = getchar()) != EOF
a[n] = &c[n]
c[n++] = ch
c[n] = 0
qsort(a, n, sizeof(char *), pstrcmp)
for i = [0, n)
if comlen(a[i], a[i+1]) > maxlen
maxlen = comlen(a[i], a[i+1])
maxi = i
printf("%.*s\n", maxlen, a[maxi]);
```

Strategy Pattern:

public abstract class Duck {

```
FlyBehavior flyBehavior;
QuackBehavior quackBehavior;
public Duck() {
}
public abstract void display();
public void performFly() {
flyBehavior.fly();
}
public void performQuack() {
quackBehavior.quack();
}
public void swim() {
System.out.println(“All ducks float, even decoys!”);
}
}

public interface FlyBehavior {
public void fly();
}
public class FlyWithWings implements FlyBehavior {
public void fly() {
System.out.println(“I’m flying!!”);
}
}
public class FlyNoWay implements FlyBehavior {
public void fly() {
System.out.println(“I can’t fly”);
}
}

public interface QuackBehavior {
public void quack();
}
public class Quack implements QuackBehavior {
public void quack() {
System.out.println(“Quack”);
}
}
public class MuteQuack implements QuackBehavior {
public void quack() {
System.out.println(“<< Silence >>”);
}
}
public class Squeak implements QuackBehavior {
public void quack() {
System.out.println(“Squeak”);
}
}
```

Observer

```
public interface Subject {
```

```
public void registerObserver(Observer o);
public void removeObserver(Observer o);
public void notifyObservers();
}
public interface Observer {
public void update(float temp, float humidity, float pressure);
}
public interface DisplayElement {
public void display();
}

public class WeatherData implements Subject {
private ArrayList observers;
private float temperature;
private float humidity;
private float pressure;
public WeatherData() {
observers = new ArrayList();
}
public void registerObserver(Observer o) {
observers.add(o);
}
public void removeObserver(Observer o) {
int i = observers.indexOf(o);
if (i >= 0) {
observers.remove(i);
}
}
public void notifyObservers() {
for (int i = 0; i < observers.size(); i++) {
Observer observer = (Observer)observers.get(i);
observer.update(temperature, humidity, pressure);
}
}
public void measurementsChanged() {
notifyObservers();
}
public void setMeasurements(float temperature, float humidity, float pressure) {
this.temperature = temperature;
this.humidity = humidity;
this.pressure = pressure;
measurementsChanged();
}
// other WeatherData methods here
}

public class WeatherStation {
public static void main(String[] args) {
WeatherData weatherData = new WeatherData();
CurrentConditionsDisplay currentDisplay =
new CurrentConditionsDisplay(weatherData);
StatisticsDisplay statisticsDisplay = new StatisticsDisplay(weatherData);
ForecastDisplay forecastDisplay = new ForecastDisplay(weatherData);
weatherData.setMeasurements(80, 65, 30.4f);
weatherData.setMeasurements(82, 70, 29.2f);
weatherData.setMeasurements(78, 90, 29.2f);
}
}
```

Factory:

```
public interface PizzaIngredientFactory {
public Dough createDough();
public Sauce createSauce();
public Cheese createCheese();
public Veggies[] createVeggies();
public Pepperoni createPepperoni();
public Clams createClam();
}

public class NYPizzaIngredientFactory implements PizzaIngredientFactory {
public Dough createDough() {
return new ThinCrustDough();
}
public Sauce createSauce() {
return new MarinaraSauce();
}
public Cheese createCheese() {
return new ReggianoCheese();
}
public Veggies[] createVeggies() {
Veggies veggies[] = { new Garlic(), new Onion(), new Mushroom(), new RedPepper() };
return veggies;
}
public Pepperoni createPepperoni() {
return new SlicedPepperoni();
}
public Clams createClam() {
return new FreshClams();
}
}

public abstract class Pizza {
String name;
Dough dough;
Sauce sauce;
Veggies veggies[];
Cheese cheese;
Pepperoni pepperoni;
Clams clam;
abstract void prepare();
void bake() {
```

```
System.out.println(“Bake for 25 minutes at 350”);
}
void cut() {
System.out.println(“Cutting the pizza into diagonal slices”);
}
void box() {
System.out.println(“Place pizza in official PizzaStore box”);
}
void setName(String name) {
this.name = name;
}
String getName() {
return name;
}
public String toString() {
// code to print pizza here
}
}

public class ClamPizza extends Pizza {
PizzaIngredientFactory ingredientFactory;
public ClamPizza(PizzaIngredientFactory ingredientFactory) {
this.ingredientFactory = ingredientFactory;
}
void prepare() {
System.out.println(“Preparing “ + name);
dough = ingredientFactory.createDough();
sauce = ingredientFactory.createSauce();
cheese = ingredientFactory.createCheese();
clam = ingredientFactory.createClam();
}
}

public class NYPizzaStore extends PizzaStore {
protected Pizza createPizza(String item) {
Pizza pizza = null;
PizzaIngredientFactory ingredientFactory =
new NYPizzaIngredientFactory();
if (item.equals(“cheese”)) {
pizza = new CheesePizza(ingredientFactory);
pizza.setName(“New York Style Cheese Pizza”);
} else if (item.equals(“veggie”)) {
pizza = new VeggiePizza(ingredientFactory);
pizza.setName(“New York Style Veggie Pizza”);
} else if (item.equals(“clam”)) {
pizza = new ClamPizza(ingredientFactory);
pizza.setName(“New York Style Clam Pizza”);
} else if (item.equals(“pepperoni”)) {
pizza = new PepperoniPizza(ingredientFactory);
pizza.setName(“New York Style Pepperoni Pizza”);
}
```

```
return pizza;
}
}
```

```
ValidIpAddressRegex = "^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-
9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])$";

ValidHostnameRegex = "^(([a-zA-Z]|[a-zA-Z][a-zA-Z0-9\-]*[a-zA-Z0-9])\.)*([A-Za-z]|[A-
Za-z][A-Za-z0-9\-]*[A-Za-z0-9])$";
```

Email:

```
[A-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]{2,4}
```

```
emailRe = /^\w+([\.-]?\w+)*@\w+([\.-
]?\w+)*\.(\w{2}|(com|net|org|edu|int|mil|gov|arpa|biz|aero|name|coop|i
nfo|pro|museum))$/
phoneRe = ^((\+\d{1,3}(-| )?\(?\d\)?(-| )?\d{1,5})|(\(?\d{2,6}\)?))(-|
)?(\d{3,4})(-| )?(\d{4})(( x| ext)\d{1,5}){0,1}$
```

矩阵乘法

```
for (i=1;i<=n1;i++)
    for ( j=1; j<=n2; ++j){
             Q[i][j]=0;
             for(k=1; k<=n1; ++k)    Q[i][j]+=M[i][k]*N[k][j];}
```

```
procedure STRASSEN(n,A,B,C);
begin
    if n=2 then MATRIX-MULTIPLY(A, B, C)
           else begin
                  将矩阵A和B依(1)式分块;
                  STRASSEN(n/2,A11,B12-B22,M1);
                  STRASSEN(n/2,A11+A12,B22,M2);
                  STRASSEN(n/2,A21+A22,B11,M3);
                  STRASSEN(n/2,A22,B21-B11,M4);
                  STRASSEN(n/2,A11+A22,B11+B22,M5);
                  STRASSEN(n/2,A12-A22,B21+B22,M6);
                  STRASSEN(n/2,A11-A21,B11+B12,M7);
```

$$C:=\begin{bmatrix} M_5+M_4-M_2+M_6 & M_1+M_2 \\ M_3+M_4 & M_5+M_1-M_3-M_7 \end{bmatrix};$$

```
                 end;
end;
void STRASSEN(int n,float A[][N],float B[][N],float C[][N])   //STRASSEN函数（递归）
{
   float A11[N][N],A12[N][N],A21[N][N],A22[N][N];
   float B11[N][N],B12[N][N],B21[N][N],B22[N][N];
   float C11[N][N],C12[N][N],C21[N][N],C22[N][N];
   float M1[N][N],M2[N][N],M3[N][N],M4[N][N],M5[N][N],M6[N][N],M7[N][N];
   float AA[N][N],BB[N][N],MM1[N][N],MM2[N][N];

   int i,j;//,x;

if (n==2)
     MATRIX_MULTIPLY(A,B,C);//按通常的矩阵乘法计算C=AB的子算法(仅做2阶)
   else
   {
     for(i=0;i<n/2;i++)                    //////////
       for(j=0;j<n/2;j++)

         {
           A11[i][j]=A[i][j];
           A12[i][j]=A[i][j+n/2];
           A21[i][j]=A[i+n/2][j];
           A22[i][j]=A[i+n/2][j+n/2];
           B11[i][j]=B[i][j];
           B12[i][j]=B[i][j+n/2];
           B21[i][j]=B[i+n/2][j];
           B22[i][j]=B[i+n/2][j+n/2];
         }                          //将矩阵A和B式分为四块


 MATRIX_SUB(n/2,B12,B22,BB);                //////////
   STRASSEN(n/2,A11,BB,M1);//M1=A11(B12-B22)

   MATRIX_ADD(n/2,A11,A12,AA);
   STRASSEN(n/2,AA,B22,M2);//M2=(A11+A12)B22

   MATRIX_ADD(n/2,A21,A22,AA);
   STRASSEN(n/2,AA,B11,M3);//M3=(A21+A22)B11

   MATRIX_SUB(n/2,B21,B11,BB);
   STRASSEN(n/2,A22,BB,M4);//M4=A22(B21-B11)

   MATRIX_ADD(n/2,A11,A22,AA);
   MATRIX_ADD(n/2,B11,B22,BB);
   STRASSEN(n/2,AA,BB,M5);//M5=(A11+A22)(B11+B22)

   MATRIX_SUB(n/2,A12,A22,AA);
   MATRIX_SUB(n/2,B21,B22,BB);
   STRASSEN(n/2,AA,BB,M6);//M6=(A12-A22)(B21+B22)

   MATRIX_SUB(n/2,A11,A21,AA);
   MATRIX_SUB(n/2,B11,B12,BB);
   STRASSEN(n/2,AA,BB,M7);//M7=(A11-A21)(B11+B12)
                              //计算M1,M2,M3,M4,M5,M6,M7（递归部分）


   MATRIX_ADD(N/2,M5,M4,MM1);                 ////////////
   MATRIX_SUB(N/2,M2,M6,MM2);
   MATRIX_SUB(N/2,MM1,MM2,C11);//C11=M5+M4-M2+M6

   MATRIX_ADD(N/2,M1,M2,C12);//C12=M1+M2

   MATRIX_ADD(N/2,M3,M4,C21);//C21=M3+M4

   MATRIX_ADD(N/2,M5,M1,MM1);
```

```
    MATRIX_ADD(N/2,M3,M7,MM2);
    MATRIX_SUB(N/2,MM1,MM2,C22);//C22=M5+M1-M3-M7

    for(i=0;i<n/2;i++)
       for(j=0;j<n/2;j++)
       {
          C[i][j]=C11[i][j];
          C[i][j+n/2]=C12[i][j];
          C[i+n/2][j]=C21[i][j];
          C[i+n/2][j+n/2]=C22[i][j];
       }                                    //计算结果送回C[N][N]

    }
```

json_pretty就是把一个一行的json string转换为容易被人阅读的格式。比如：

输入：{“id”:"id-123","woe_id":[123,456,789],"attribute":{"title":"a","desc":"b"}}

```
输出：{
       “id”:"id-123",
       "woe_id":[123,456,789],
       "attribute":{
          "title":"a",
          "desc":"b"
       }
     }
```

```
public class JsonNode {
   String nodeName;
   Value nodeValue;
}

public class StringValue extends Value {
   String value;
}

public class NodeValue extends Value {
   JsonNode[] value;
}
```

然后分拆的代码：

```
private JsonNode parseNode(String jsonString,
    int startIndexInclusive, int endIndexExclusive) {

    assert (startIndexInclusive >= 0);
    assert (endIndexExclusive > startIndexInclusive);
    assert (endIndexInclusive <= jsonString.length());

    // find the first semicolon
    int sc = jsonString.indexOf(':', startIndex);
    if (sc < 0 || sc >= endIndex)
        throw new InvalidFormatException();

    String name = jsonString.subString(startIndex, sc);
    String value = jsonString.subString(sc + 1, endIndex);
```

```
     Value nodeValue = value.startWith("{") ?
         new NodeValue(parse(value)) : new StringValue(value);

     return new JsonNode(name, nodeValue);
}

public JsonNode[] parse(String jsonString) {
    List<JsonNode> buffer = new ArrayList<JsonNode>();

    for (int i = 0; i < jsonString.length(); i++) {
         char ch = jsonString.charAt(i);
         boolean switch = (ch == '{');

         int j = switch ? i + 1 : i;

         for (; j < jsonString.length(); j++) {
              if (switch && jsonString.charAt(j) == '}') ||
                  jsonString.charAt(j) == ',')
                  break;
              }
         }

         JsonNode node = parseNode(jsonString, i, j);
         buffer.add(node);

         for (i = j; jsonString.charAt(i) != ','; i++)
             ;
    }

    return buffer.toArray(new JsonNode[buffer.size()]);
}
```

打印的代码就好写多了：

```
public void print(PrintStream ps, JsonNode[] nodes, int indent) {
     for (int i = 0; i < nodes.length; i++) {
          JsonNode node = nodes[i];

          printIndent(indent);
          ps.print(node.getName());
          ps.print(":");

          if (node.getValue() instanceof StringValue) {
               ps.print(node.getValue());
               if (i < nodes.length - 1)
                   ps.print(",\n");
          } else if (node.getValue() instanceof NodeValue) {
               JsonNode[] value = ((NodeValue)node.getValue()).getValue();
               ps.print("{\n");
               print(ps, value, indent + 4);
               printIndent(indent);
               ps.print("}\n");
          }
     }
}
```

然后问我给定一个FaceBook log file，100 billion行，每一行记录含“timestamp user_id visited_page”，找到top 10 最长出现的三连串访问模式。比如user先后访问了页面a,b,a，那么就形成一个模式a->b->a。但是记录没有按照timestamp排好序！

我的答案是，1）brute force：对整个文件先对timestamp再对user id排序，然后建hash表，读入log文件，对每一个模式计数。 2）map-reduce：将每一个user id对应的记录读到单独的机器上--map，接着对timestamp排序建hash表，这样效率高些，然后--reduce，将intermediate结果对每个模式key累加，计算最后结果。然后因为只需要top 10，可以用一个10个元素的min heap维护当前top 10。

给定int read(char *buffer, int size) api，写出int readline(char * buffer, int size)code。举例说，输入一个字符串“abcd\nefgh”，read(buffer,3)返回3，buffer=abc，同时指针指向字符"d"。read(buffer,7)返回7，buffer=abcd\nef，同时指针指向字符"g"。返回值是实际读取的字符数。要求readline(buffer,3)返回3，buffer=abc，同时指针指向字符"d"。readline(buffer,7)返回4，buffer=abcd，同时指针指向字符"e"

红绿灯:

```
while (true) {
    setBlue(direction1);
    setRed(direction2);

    sleep(30 * 1000);

    setRed(direction1);
    setBlue(direction2);

    sleep(30 * 1000);
}
```

Number of Sum

```
void numberOfSum(int sum, vector<int> v, int last)
{
   if(sum == 0)
   {
      for (int i = 0; i < v.size(); i++)
      {
         cout<<v[i]<<",";
      }
      cout<<endl;
      return;
   }
   for (int i = 1; i <= sum; i++)
   {
      if(sum - i >= 0 && i >= last)
```

```
    {
      vector<int> temp = v;
      temp.push_back(i);
      numberOfSum(sum-i,temp,i);
    }
  }
}



void dijkstra(int start)
{
  priority_queue<pair<int,int> > queue;
  pair <int,int> nodotmp;
  int i, j;

  for (int i=1; i<=total; i++) {
    distances[i] = MAXINT;
    father[i] = -1;
    visit[i] = false;
  }

  distances[start] = 0;
  queue.push(pair <int,int> (distances[start], start));

  while(!queue.empty()) {
    nodotmp = queue.top();
    queue.pop();
    i = nodotmp.second;
    if (!visit[i]) {
      visit[i] = true;
      for (j = 1; j<=total; j++)
        if (!visit[j] && graph[i][j] > 0 && distances[i] + graph[i][j] < distances[j]) {
          distances[j] = distances[i] + graph[i][j];
          father[j] = i;
          queue.push(pair <int,int>(-distances[j], j));
        }
    }
  }
}


void getPath(int end) {
  cout << end << " ";
  while (father[end]!= -1) {
    cout << father[end] << " ";
    end = father[end];
  }
  cout << endl;
}

int main()
{
   int a, b, c;
   int tedges;
   memset(graph, 0, sizeof(graph));
   cin >> total >> tedges;
   for (int i=0; i<tedges; i++) {
     cin >> a >> b >> c;
     graph[a][b] = c;
   }
   for(int i=1; i<=total; i++) {
      for(int j=1; j<=total; j++)
        printf("%d ", graph[i][j]);
      printf("\n");
   }
   dijkstra(1);
   getPath(3);


   return 0;
}
```

Moving-window maximum.

Input: A long array A[], and a window width w
Output: An array B[],B[i] is the maximum value of from A[i] to A[i+w-1]
Requirement: find a good optimal to get B[i]
I can think of two solutions

First solution:
if(A[i] = !B[i-1]) B[i] = B[i-1]
else B[i] = max{A[i],...A[i+w-1]}

Second solution(this one is from one of my friend)
Use minHeap to store w number, each time when a new number A[i] enters, age out the oldest one, and use O(logW) time to get max value of A[i-w+1] to A[i].
O(n) algorithm:

```
    public Integer[] getMaxInSlideWindow(Integer[] A, Integer w) {
        // invalid input
        if (A == null || w <= 0 || A.length - w + 1 <= 0)
            return null;

        Integer[] B = new Integer[A.length - w + 1];

        // auxiliary queue that is sorted in descending order
        List<Integer> q = new LinkedList<Integer>();

        for (int i = 0; i < A.length; i++) {
            // enqueue. Remove those smaller values
            int data = A[i];
            while (!q.isEmpty() && q.get(q.size() - 1) < data) {
                q.remove(q.size() - 1);
            }
            q.add(data);

            if (i < w - 1)
                continue;

            // dequeue. If the current number is the maximum. Also remove it
            // from the queue
            Integer max = q.get(0);
            B[i - w + 1] = max;
            if (A[i - w + 1] == max)
                q.remove(0);
        }

        return B;
    }
```

1. 给定两个整数数组，找出同时出现在两个数组中的整数。

   2.如果不允许O(n)的空间，怎么办？

   3.如果数组太大无法装入内存，怎么办？

   4.如何测试？

   5.如果一个web-based系统中某个网页crashed，如何检测

what does class Derived : private Base mean?
what does class Derived : protected Base mean?
why initialize member variables in initializer list of constructor, rather than body of constructor?
discuss different facets of polymorphism in C++ - dynamic, static. Define polymorphism clearly.
what are the adapter, builder, command, singleton patterns in C++?
what happens when you remove elements from a vector STL container, compare vector before vs after removal.
what is the 1 thing you can have in a class declaration that you can never encounter in a union declaration?
what are the 4 scenarios where the copy constructor comes into play?
what are the 2 main categories of signals in UNIX?
difference between re-entrant code vs. thread-safe code? How about async thread-safe?
big endian byte order test inside the body of a function
mutex vs. semaphore difference?
fastest algorithms to find common elements of 2 distinct arrays? Give complexity in big O notation
find the smallest K elements in a set of N > K elements, describe complexity of algorithm
effect of database index in SQL databases on performance (i.e. insertion/lookups). Bulk loading considerations.
describe a good thread pool implementation in C++ (hint: stay away from C idioms to do well on this question), look at Boost's bind class
describe functors in C++

nterviewer 1: 写一个函数validate XML tags (类似parentheses matching). 我用了stack. 然后测试这个函数。答完之后倒是聊了很多关于这个组的工作。

interviewer 2: lunch interview. 我选择take out, 所以在办公室里边吃边聊。把一个BFS转成doubly LL in BFS fashion. 我实现了BFS traversal但是发现pointer relinking很难。结果面试官说他也不确定能否实现。于是我们就把BFS都放到一个array里，然后relink pointers. 第二个问题，有一个洗牌程序，如何测试是不是真的随机。

interviewer 3: Why microsoft? Why SDET? What programs I’ve written? What is the most challenging? 说了一些PhD research. 最后出题：设计一个airplane control system. 如何测试。

interviewr 4, Test Lead: Why microsoft? Why SDET? 让我介绍了一下我简历上的所有experiences. 问题: 一个单线程程序 looping over a million records, 一个双线程程序，每个线程looping over 500,000 records, respectively. Which one is faster? 在什么情况下那个双线程程序的效率会下降？问题：有一环数字，每隔一个数字，划掉一个数字。如此循环，直到所有数字都划掉。写程序实现，随便什么数据结构都行。（我用了doubly LL，先写了程序。然后立刻debug, 处理了special cases, such as when there are two numbers left and there is only one number left).

interview 5, Dev Lead: 如何测试一个程序？看不看有关Software engineer的杂志/周刊？问题1: 估计一下一个map system(such as google map)需要多少存储。他需要实际数字，所以每一步都要参考实际情况。比如说地球的半径（6400KM), 而后计算表面积。然后，假设最小单位是20米乘20米，估算这个最小单位显示时为100X100pixel, 每个pixel(R/G/B)占3bytes, 所以这个最小单位占30KB, 压缩后大约1KB..... 我算下来大约1.2PB = 1200TB. 然后再考虑zoom，支持多少级zoom. 不过zoom out之后，空间小很多，所以总和还是在1.2PB左右。问题2: 一个已排序但是有重复的int array. rotated. 写一个函数，return 原来index=0现在的位置。比如4,4,5,6,6,1,1,2,3, 函数returns 5 . 如果数组是2,2,2,2,2,2,2, 则return 0. 对于没有重复的情况，可以实现logN, 但是现在只能实现N. 这个interviewer对coding非常picky, 不能有一点pseudo code, 而且对程序要求很高（那种为追求效率，宁可牺牲可读性的程度

电面1：问了Java的各种基本概念，Java 里面int多大，怎么知道超过范围了， 链表检测loop，Java 里面的linkedlist检测loop， 然后是一个brain teaser, 和扔鸡蛋问题差不多。这个没回答上来。

电面2：三道题目，都很简单，第一道是链表中倒数第n个node是什么，第二道题目是数组中只有一个数字出现了一次，其他出现两次，找出那个数字。
第三道题目就是设计一个chess。

然后就给on site了，on site也不太难。所以感觉运气还不错，没有网上看到的那些变态题目。

具体的顺序忘了，不过问过如下的题目：

设计数据库的表储存网上购物时候的order
给了n个线段，然后知道他们的开始结束的坐标，返回有多少条线段相交。关键就是写个代码判断两条线段有没有相交。
设计outlook的calender（这个由于没怎么用过outlook， 回答的很烂，完全不是对方期望的答案）
两种方法写斐波那契数列。
两种方法写出给定一个字符集合的所有子集。
知道n个雇员的住址坐标，然后知道办公地点的坐标，有一辆班车要接送所有的雇员，停靠5站。优化公交车站点。
判断二叉树是否平衡，怎样维护（这个就没让写代码了）。插入一个二叉树节点worst case是什么。
一个二维数组，从左到右，从上到下都是增序， 找出这个二维数组是否包含某个数。
吃午饭的时候还被问到有关动态维护前100个买的最多的商品的问题，只不过商品信息不能全部载入内存中。然后每一次交易发生，都会写一个entry到log file中，怎么维护。后来又改成有很多服务器储存商品的信息，怎么维护。

1. 两个文件里面存着行程安排，开始时间以及结束时间，设计算法找conflicts

   2. How would you find the first unique url among the millions of url available?

   3. How to sort an array of only three possible values。

4. 大数乘法。我是用linked list做的。是不是一般都用array做？ 那位给个简洁的code.

```
BigInt BigInt::multiply(const BigInt &b) const
{
  BigInt answer;
  BigInt temp;

  int carry = 0;
  for (int i = 0; i < b.numDigits; i++)
  {
    temp.numDigits = this->numDigits + 1;
    int down = b.digits[i];

    for (int j = 0; j <= this->numDigits; j++)
    {
      int up = (j < this->numDigits) ? this->digits[j] : 0;
      int tempDigit = up * down + carry;

      temp.digits[j] = tempDigit % 10;
      carry = tempDigit / 10;
    }

    temp.shiftDigits(i);
    temp.zeroJustify();
    answer = answer.add(temp);
  }
  return answer;
}
```

1 load all schedule to a arraylist/vector, 所有行程按start time sort（根据要求选sort algrithm O(n)(radix/counting) or O(nlogn)）, 然后从头走一遍 ending time 跟后面的 start time 比较，如果有 conflict 就做mark 并且用老的endtime跟当前
endtime比较 做 更新,
2 harsh
3 dutch national flag
4 这种题看着头晕，需要细心和耐心

1. 谈一下不同数据结构的优缺点。
2. 一个大文本文件里有电话号码。每行至多有一个号码。How do you process and return the total number of phone numbers. (in command line, use grep and wc)
3. A generic tree. how to print out nodes by level (one lever a line)
说了pseudo code, 要求电面后email给他。
4. A database application is slow. How do you investigate the problem and how to improve it.

Second 电面:

1. 写一个函数，输出一个整数里1-bit的数目。比如CountOneBits(7)应返回3。
2. 网页很慢。找出可能原因。（和一面的最后一个问题差不多）
3. 下面statements的区别是什么？ 接着问了关于constructor的问题(copy), shallow vs. deep copy.

```
Class A;
A a = u;
A a(u);
A a();
A a;
```

4. What is virtual function and polymorphism?
5. What is reflection?
6. A large time-stamped log files, 怎么找一个时间范围内logs? (grep)
7. What is the difference between hashing and encryption?

Third 电面 （口音比较重的老印。不少地方听不明白，只能不断叫他慢速重复。他人倒是不错，很客气）:

1. 谈了一下phd research
2. 一个文件里一百万个数字。怎样找到最小的100个? (Use min-heap of size 100)
Ask about complexity (nlogn)
3. 写函数 输出Fibonacci numbers in normal sequence (no loop allowed, use 递归,
cannot compute Fibonacci number using F(X) = F(X-1) + F(X-2)) 我用了两个static variables. 程序写好念给他听。他运行了一下说pass.

4. 写函数 输出Fibonacci numbers in reversed sequence, 和上题同样的限制。电话里没有想出来，总觉得需要一个stack的数据结构，才能实现反序输出。他给我30分钟让我电话后写了程序email他。最后我还是用了一个辅助的数据结构。他reply我，给了idea. 实际上还是用两个static variable. 在recursion中计算，在foo(0)里面输出F(X), 然后在recursion返回中在还原出F(X-1), F(X-2) …. 直到F(1), F(0).

Onsite （because of NDA, 只能说的抽象一点）:

组A的dev manager:

1. OO design该公司的一个system， 设计的时候，使用patterns. 这个问题初听很难，所以我就不断地问问题。最后把问题简化到一个可以操作的程度。这样设计起来方便不少。
2. 如何在文件里找电话号。(他们的确喜欢grep)

Recruiter

组A的SDE:

1. 谈research
2. 给一个int array, 输出一个随机permutation. (实际上就是random shuffle) 用了reservoir sampling. 然后演示了一下1,2,3的六种输入可能。
3. difference between i++ and ++i
4. what is virtual function? why C++ makes non-virtual default? (speed tradeoff)

组A的SDET:

1. 写函数: given an integer, 判断这个integer, in binary format, 是不是一个回文结构。
2. 一个binary tree of integers. serialize and then desialize.

非面试组来的面试官:

1. 谈research
2. 公司网站上有一个错误的电话号码。怎么找出来。一开始回答得有些不得要领。后来说到要web crawl一下公司的domain, 他似乎有兴趣。于是就用bfs traverse, 用hash table来防止infinite loop. 对于每个page, 自然还是grep来找。

组B的dev manager(lunch interview):

没问任何技术问题。基本在聊天。问我，什么才是好的程序？怎么样comment程序？很多时候在说他们组的东西。

该公司正在从C++到java转型。但面试我的人似乎都对C++比较熟悉。至于用什么语言回答问题完全不重要。可能我说自己C++比较熟，所以他们没有问java specific的问题。

第一次电面， 白人
介绍自己的research
c++ questions: interface, abstract class, polymorphism
find nth to last node in linked list
design card game, extend to a specific game

第二次电面， 印度人，有一点口音
比较变态，迟到十分钟，说电话号码搞错了，上来就说了一堆迟到的理由
让我介绍research

如果我是经理，有一个大的项目，怎么把这个大项目分成小项目分给手下人做，要high level, 扯软件设计的思想，user case?我没听说过。

Amazon 有一堆数据，量很大，要求设计一个系统，搭建平台，如何用machine learning, data mining建模

面到这儿我觉得被打了个措手不及，觉得应该挂了，45分钟到了，他又出两道题，一个小时结束
题一： 两个array找common numbers
题二： 亚麻有一个大的array里面是所有书的作者， 给定长度N,要求返回随机的t个作者，要求概率相同。 follow up, 要求每天返回随机t个作者，而且前一天返回的作者不能出现在后一天的名单中

Phone 1:
1.提出尽可能多的方法使一个method可以返回多个不同type的值
2.reverse string
  比如 "I have a dream" -> "dream a have I"
3.判断一个binary tree是不是对称的

Phone 2:
1.给a list of number，返回前top K个（内存足够怎么做，内存不够怎么做）
2.OOD 电梯
3.找两个链表的交集

Onsite 6轮 1轮HR 1轮午餐 4轮技术 （亚马逊网络服务组）
签了保密协议，希望不要被抓到T_T
1.设计个电话本 可以用那些数据结构
  我说suffix tree, 哈希表
  问了这两种方法的比较，还考了suffix tree的插入，
2.问research, OOD 交通灯系统
3.写函数算一个整数的阶层 n!
  又问了n很大，怎么办？
  比如99%的n都在400000-900000之间，怎么提高函数的执行速度
4.给一个数组和一个数n，找出数组中的所有的对和等于n的
5.给手机键盘，给定某个按键序列比如‘1489’，返回这个按键序列生成的所有的正确的单词

```
typedef struct range
{
    int begin;
    int end;
} Range;


Range shortestSubstring(const char* str, int strLen, const char* characters,
int charCount)
{
    int* needToFind=new int[256];
    int* hasFound=new int[256];

    for(int i=0;i<256;i++)
    {
        needToFind[i]=0;
        hasFound[i]=0;
    }

    for(int i=0;i<charCount;i++)
    {
        int index=(int)characters[i];
        needToFind[index]++;
    }

    int count=0;
    // count is used to judge if the range satisfies the requirement
    // the criterion is count==charCount

    int begin=0;
    int end=-1;
    int index=0;

    Range minRange;
    minRange.begin=-1;
    minRange.end=-1;

    int minLength=strLen+1;
    int length=0;

    while(end+1<strLen)
    {
        end++;

        index=(int)str[end];
        if(needToFind[index]>0)
        {
            hasFound[index]++;
            if(hasFound[index]<=needToFind[index])
            {
```

```
            count++;
        }
    }


    while(begin<end)
    {
        index=(int)str[begin];
        if(needToFind[index]==0)
        {
            begin++;
        }
        else if(hasFound[index]>needToFind[index])
        {
            begin++;
            hasFound[index]--;
        }
        else
        {
            break;
        }
    }


    // cout<<"begin:"<<begin<<", end:"<<end<<", count:"<<count<<endl;

    if(count==charCount)
    {
        length=end-begin+1;
        if(length<minLength)
        {
            minLength=length;
            minRange.begin=begin;
            minRange.end=end;
        }
    }

  }

  return minRange;
}

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MYPORT 1234    // the port users will be connecting to

#define BACKLOG 5     // how many pending connections queue will hold
```

```
#define BUF_SIZE 200

int fd_A[BACKLOG];    // accepted connection fd
int conn_amount;    // current connection amount

void showclient()
{
    int i;
    printf("client amount: %d\n", conn_amount);
    for (i = 0; i < BACKLOG; i++) {
        printf("[%d]:%d  ", i, fd_A[i]);
    }
    printf("\n\n");
}

int main(void)
{
    int sock_fd, new_fd;  // listen on sock_fd, new connection on new_fd
    struct sockaddr_in server_addr;    // server address information
    struct sockaddr_in client_addr; // connector's address information
    socklen_t sin_size;
    int yes = 1;
    char buf[BUF_SIZE];
    int ret;
    int i;

    if ((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }

    if (setsockopt(sock_fd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) == -
1) {
        perror("setsockopt");
        exit(1);
    }

    server_addr.sin_family = AF_INET;         // host byte order
    server_addr.sin_port = htons(MYPORT);     // short, network byte order
    server_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my I
P
    memset(server_addr.sin_zero, '\0', sizeof(server_addr.sin_zero));

    if (bind(sock_fd, (struct sockaddr *)&server_addr, sizeof(server_addr)) =
= -1) {
        perror("bind");
        exit(1);
    }

    if (listen(sock_fd, BACKLOG) == -1) {
        perror("listen");
        exit(1);
    }

    printf("listen port %d\n", MYPORT);

    fd_set fdsr;
    int maxsock;
    struct timeval tv;

    conn_amount = 0;
    sin_size = sizeof(client_addr);
    maxsock = sock_fd;
    while (1) {
        // initialize file descriptor set
        FD_ZERO(&fdsr);
        FD_SET(sock_fd, &fdsr);

        // timeout setting
        tv.tv_sec = 30;
        tv.tv_usec = 0;

        // add active connection to fd set
        for (i = 0; i < BACKLOG; i++) {
            if (fd_A[i] != 0) {
                FD_SET(fd_A[i], &fdsr);
            }
        }

        ret = select(maxsock + 1, &fdsr, NULL, NULL, &tv);
        if (ret < 0) {
            perror("select");
            break;
        } else if (ret == 0) {
            printf("timeout\n");
            continue;
        }

        // check every fd in the set
        for (i = 0; i < conn_amount; i++) {
            if (FD_ISSET(fd_A[i], &fdsr)) {
                ret = recv(fd_A[i], buf, sizeof(buf), 0);
                if (ret <= 0) {        // client close
                    printf("client[%d] close\n", i);
                    close(fd_A[i]);
                    FD_CLR(fd_A[i], &fdsr);
                    fd_A[i] = 0;
                } else {        // receive data
                    if (ret < BUF_SIZE)
                        memset(&buf[ret], '\0', 1);
                    printf("client[%d] send:%s\n", i, buf);
                }
            }
        }

        // check whether a new connection comes
        if (FD_ISSET(sock_fd, &fdsr)) {
            new_fd = accept(sock_fd, (struct sockaddr *)&client_addr, &sin_si
ze);
            if (new_fd <= 0) {
                perror("accept");
                continue;
            }
            // add to fd queue
            if (conn_amount < BACKLOG) {
                fd_A[conn_amount++] = new_fd;
                printf("new connection client[%d] %s:%d\n", conn_amount,
                        inet_ntoa(client_addr.sin_addr), ntohs(client_addr.si
n_port));
                if (new_fd > maxsock)
                    maxsock = new_fd;
            }
            else {
                printf("max connections arrive, exit\n");
                send(new_fd, "bye", 4, 0);
                close(new_fd);
                break;
            }
        }
        showclient();
    }

    // close other connections
    for (i = 0; i < BACKLOG; i++) {
        if (fd_A[i] != 0) {
            close(fd_A[i]);
        }
    }

    exit(0);
}
```

1. 二叉树中给定一个节点，查找按照中序遍历顺序它的后继节点，要求写代码，并给出复杂度；二叉树中查找中序遍历顺序中的第k个节点，如果每个节点都添加了子树中节点个数这个变量，如何在插入、删除和旋转时更新这个值（旋转是为了保证logn的复杂度而要使二叉树保持平衡）。

   二、C++概念题，包括虚函数、多继承、私有的构造、析构函数、重载的new运算符等；以前的project问题；开放性问题，跟网络有关，包括了分组交换、拥塞控制、流控制、多播等等知识点；最后问了一个编程题，跟quad tree有关，不太常见，但不是很难，我觉得考查了函数的递归。

   三、一道编程题，大意是给定一个类read1，它有一个函数read4096，每次调用它可以从文件中读取4K个字节，同时移动文件指针4K个位置（若文件中剩余数据不足4K，则读取剩下的所有数据），这个函数返回实际读取的字节数，int型；要求实现另一个类read2中的一个函数read，它有一个参数int n_byte，这个函数可以从文件中读取由n_byte指定的字节数，同样返回实际读取的字节数；然后又给出一个函数reset，它可以将文件指针重置到起始位置，要求实现read2中的另一个函数seek，有一个参数int pos，它可以将缓冲区的指针移动到第pos个字节的位置，返回实际指针移动到的位置。可以在read2中添加任意变量来完成这两个函数。
   这道题也不难，需要注意代码的一些细节，比如循环的终止条件、特殊的输入等。

第一轮，问了一下自己觉得最有意思的项目。然后就是3个题：有一个很大的Log文件，记录了每个用户点击网页的时间，问怎么找到最常见的3连击；有两个很大的文件，文件里每行都是string，问怎么找到重复的；找一个无序数组的第k大元素。

第二轮，很多基本的问题，比如什么是hash,怎么处理冲突；然后什么是encapsulation，什么是inode。大多是基本概念。然后问了个程序题，怎么验证一个数是不是素数。
最后考了一个OOD，那个电梯的题目。

第三轮，两个进程之间有多少种方式可以互相通讯（尽量说，不要管效率）。然后问了问怎么处理race condition。接着就是验证一个二叉树是不是BST。然后问了一个设计题，题目描述太复杂了。。很难复述。。然后俺就跟面试官聊啊聊，后来才发现他想要一个多态的设计。

1. 两个圆在什么条件下相交？
   2. m*n的矩阵in place rotation?

　看见阿三我心就凉了半截。年纪大了，反应慢，算算术吭哧吭哧，第一题就捣持了半天。第二题就别提了，吭哧到最后，也就是讲了讲这题有什么corner case,难点在哪，说如果换做n*n的就简单多了。三男非常满足的在一边幸灾乐祸的从头笑到尾，把我写的任何一个字，画的图，说得任何一句话都恨不得要记下来。后来他让我写个不是in place的了事。回来我google半天，也没有找到这道题在任何地方被提起和讨论过。我后来discussion的时候问他答案是什么，他也不说，就说这不是个straightforward的问题，说我们主要是看你解决问题的思路，我觉得you are doing quite well, don't worry about this. 也许是看自己第一个面我，折磨成那样，良心发现了安慰一下。

白男：
  1. n个城市之间的距离要把都存下来，怎么存最省空间？
  2. 前序中序重建树。

　其实这个白男非常拽，也有点岁数了，估计35超上。但这个说的和写得都让他很满意，应该是最没问题的一个。最后还剩10多分钟，随便聊了聊。

国女：

0. 也许也有个什么warmup, 我忘了。
1. 64 bit的integer,怎么数里面1的个数？
   followup: 要是多次使用你怎么办？你不觉得要用空间的太多了吗，怎么办？
2. p*q的matrix,从左下到右上路径数？
   followup: 你这个算的会有什么问题？你怎么解决？
   followup: matrix中有障碍呢？（其实我没有感觉到时间过的很快，这个没有code完，她让我说了说算法了事）

看到国女我喜极生悲，简单的coding被她揪出来bug，不过我不知道她是不是非常nice的没有记下。要是她能看到这个帖子，我很想说声大姐你很漂亮！

三女：

1. m长的array，长度为k的sliding window,求每次slide一下window里的最大值。然后问test cases.
2. 你对网络了解吗？（不了解。）好吧那多线程呢？（还行吧）于是问mutex, semaphore概念，出了个多线程的题，一点一点的深究设计。

三女很认真，虽然跟三男风格大不一样，几乎不记笔记，估计全记脑子里了，但是她非常认真的在纸上画，试图抓我bug，我面对三女当然谨慎了，没有给她抓到bug。多线程的设计具体确实不记得了，最后问到一个地方，我愣了20秒没答上来，她心满意足的发表结束陈词：我们就是想让你能有点东西可以想的，要是你所有问题都答上来了，说明我们interviewer没有do a good job. 我觉得她很有诚意，应该也没有写太多坏话。

1. 给一段C程序（汗，C我真没仔细学过）看有什么问题，具体的忘了，好像是关于函数里的char*出了函数就有问题的事。
2. 从m长的array中随机取里面的n个，怎么做？数学推导？好像还有另外一个也是sampling啥的，忘了。
3. n-bit的integer，打出所有有k个bit被set的数，你这个复杂度多少？怎么提高？还是不够高，怎么办？
4. 打印power set。
5. 多线程细节，hashtable细节。
6. 用Java写一个iterator,满足一些要求，细节不记得了，有一点tricky。
7. 给我讲STL里的unique函数是干什么的，让实现，怎么提高效率？
8. 读一个什么文件，问看那些数据我想到些什么，然后写程序求最大，最小，average之类的。

```
for n = 0 to N - 2
    for m = n + 1 to N - 1
        swap A(n,m) with A(m,n
```

```
for each length>1 cycle C of the permutation
    pick a starting address s in C
    let D = data at s
    let x = predecessor of s in the cycle
    while x ≠ s
        move data from x to successor of x
        let x = predecessor of x
    move data from D to successor of s
```

上来就一个问题,设计一个网上预定飞机票的系统

讨论了50分钟左右,弄的很细,每个class里有啥variable,啥method,都要说. 比如
Controller class谁去call, 怎么用Controller class. 还有内存怎么寸(不用数据库,
全存memory里),用什么数据结构,(他提出hashtable),什么作为key,什么作为value
一开始还好,后来纠结在一个如何给用户一个指定日期的航班信息,因为我没有存日期,
最后时间快到了,他让我把想好的设计发邮件给他...

面完就忽然想出来了, 不过觉得面的一般,老被烙到处印牵着问
今天都周末了还没消息,准备move on..

2. Boston的一个大公司,老板转给我的邮件,发信给老板说招人
马上给我安排了电面,结果发现是project manager
1小时左右,都是算法题
一个问apache一个什么log里面如何找前10个频率最高的ip
然后说如果前k个呢

第二题就那啥了,说从n个数里取m个随即数,前后不重复
我说了reject sampling, shuffle algorithm, 提到了reservior sampling,他说他没
听说过reservior sampling

然后就悲剧了,他要我优化shuffle那个算法,说现在是一个连续的整数空间,要keep
track of the holes, 不要象shuffle那样swap来maintain O(n)的space
我讲了一个O(m^2) time的, 用LL存holes, 还要求被优化, 后来我犯晕了, 他说我想好
后给他邮件.....
我想了一个下午也没想出来怎么不用linear data structure 去maintain那个holes
list, 发信给他说没想出来, 拜托告诉我solution, 另外副了之前说的3个算法

他给我安排了2nd 店面,到现在也没告诉我算法,还说那个reservior sampling不错, 时
间O(n)小于O(m^2),靠,不废话吗,害我白想一下午! 八成自己回家发现自称的优化的结
果不存在..

和我专业（machine learning）相关的很多问题，谈得比较细。
记得的算法：
given one array，找出两个数的和为给定的数
给一个string，求所有的permutation。
OOD 设计graph类
BST is valid
各种sort比较
linkedlist 和hashmap的相关细节
graph bfs

还有几个算法题我忘了，都是经典的。除了个别的刚开始想复杂了，其他的基本给的经
典答案。

第三面刚完，大意了，没从名字上判断出是老印，交流是相当有问题。而且一上来态度
就和我欠了他钱似的。。。跟他的面试记录等我休息会儿慢慢补上。
就面了一道算法题。Given a stream of integer.given one integer K as the
window size, compute the average of the numbers in the latest window.
for example, 1 2 3 4 with K=2
output 3.5
以前没见过，不过还是很快给出啦算法，计算accumulated sums, maintain 一个K+1 size的

circular array to store the latest sums. 这样计算latest average的时候就是
constant time的时间复杂度。
老印态度立马好转（之前听不懂他说话，他就有点冲），说他挺满意的。我以为pass了，结果他让开
始写code，交流完全不畅，丫态度变得相当不好了。他只给了3分钟就说你写完了吧。。

解释一下research中用过的machine learning算法
有一个项目中做了一个数据库，把数据库结构画出来，解释各个entity之间的关系

join的种类，区别
sql用的是哪种(mysql之类)

difference between struct and class

features of object oriented design
give examples；how did you used OOD in your research
give the definition of the classes used in your research project
virtual function; pure virtual function; abstract class

for base/derived classes,
destructor virtual or not virtual, why?
constructor virtual or not virtual, why?

int a; static int b;
what is difference?
where do they store?

new产生的对象存在哪里

what is semophore?
multi thread里面你用过哪种(pthread)
pthread怎么定义mutex(写出命令)

lock and unlock
deadlock
give the example of deadlock
and how to avoid or deal with deadlock/circular wait
(answer: 同时lock所有需要的资源)
research里面的哪个项目用了multi thread
有没有遇到过dead lock，如何解决

multi-threading定义
how to synchronize

how to communication with each other between multi threads.

pass by reference, pass by pointer, difference of them
void foo (int* p)
void foo (int& p)
哪个是pointer，哪个是reference，在这个function里的区别

tell me about all kinds of sort algorithms you know

列三个，讲一下大概怎么实现(写伪代码)
有几千个文件，要按照文件名sort，应该用哪种sort

list all the data structure you know
大概讲一下定义

coding:
two strings A and B
1.check if A is B's anagram(忘了怎么拼了，就是字母相同顺序不同的意思)
2. check if all the characters in A appear in B (不考虑重复的字母)
3. 如果考虑重复的字母，比如a在A里面出现两次，在B里也应该出现两次，2的程序怎么改
hashtable在C++还是JAVA里是定义好的可以直接用？

dynamic allocated memory，最后要怎么处理(delete)
怎么保证分配的空间一定被delete掉？有什么工具可以检查(老实说不知道)

memory leak是什么？举例说明
怎么处理

举出三种design pattern来
给出代码
你的research project中用过哪几种
为什么用这几种

如果一个class，同一个功能在不同的地区有不同的method，并且参数类型也不一样
怎么办
用哪一种design pattern

design pattern的定义
为什么大家要用design pattern

TCP/UDP，定义，区别
sliding window
IPv4/IPv6的定义，区别，知不知道IPV4几个星期前用完了

让设计一个游戏，楼主我不会玩这个游戏，就算了
改设计了个别的，忘了...

又想起来一题，补充一下：
process和thread的定义，区别

1. ８瓶水，一瓶有毒。老鼠喝了毒水第二天会死去。问用几只老鼠能尽快的找到毒水。
２。二叉树给两nodes找最近公共祖先
３。给n＊m的字符矩阵。然后给你一个字符串。问是否可以在矩阵中找到他的track。track是指从其
中一个符出发，可以向四周走，可以重复，可以回头

1. Suppose we have a stream of web clicks. Now you need to provide at any time the count of web clicks during the past 1 minute. To be specific, you get informed whenever a web click happens. You need to provide a function " getCount()" such that once called, return the count of clicks during the past 1 minute. The solution will be both constant in time and space.

2。 N * N MATRIX, 只有一行完全是 0, 其他行有 0 也有 1, 怎么最快找到完全是 0 的那一行. 平均O()和 WORST CASE O() 是多少?

一个奥运会网页，上面要显示每个国家拿了几块奖牌，金牌银牌铜牌各多少块
如果拿了新的奖牌，要更新网页
问怎么设计class
要求写代码

2. 如何设计一个停车场？停车场里有n个车位，每个车位离入口有个距离，当一辆新车进来以后，给他安排最近的车位。
3.写一个程序，计算给定字符串中单词的数目，如何测试？

第二个人，
1.最挑战的project
2.设计一个电话本，如果内存不够，你用binary tree还是hashtable? (我答的binary search tree，不知道对否？）如果想把电话本里面所有的item按字母顺序输出，如何实现（binary tree 和hashtable都要回答）？
3.如何寻找文件中所有的电话号码？123-456-7890
4.有一个array,每个元素是一个string,给定一个string s,查询s的reverse string是否出现在array当中。

Phone 1 别的组的老美
两个数组求交集。如果已经排好序了，一个数组很大，一个很小怎么办。如果数组都很大，内存放不下，怎么办。
设计扑克牌。扑克牌shuffle算法。
两个整数，需要多少步才能把一个数的二进制表达转换到另一个数的二进制表达。（XOR后数1）
Phone 2 本组的印裔
设计LRU Cache, 然后讨论多线程访问Cache的问题。面完后实现Cache发代码给他。
Onsite见了7个人，每个人45分钟，连轴转。上午10点半进building, 下午4点出来
Onsite 1 很Nice的老美
讨论设计web crawler， coding BFS， 讨论多线程处理crawler等。
Onsite 2 印裔
OOD机场air traffic control system.
Onsite 3 听口音像感觉英国人
跟经理吃饭，common questions, 说一下你跟coworker有disagreement，怎么处理。问他问题。
Onsite 4 亚裔凶GG
一个网站，如果有大量访问，怎么设计结构。这个问题太开放了，没有什么经验，答的不好。。。
给个矩阵，每个格子是一个字母，每次可走8个方向，输出valid word. 假如有个字典可以判断每个word是否valid. 把矩阵看成图，其实就是个简单的DFS。当时被上一个问题搞的有点晕，写了个iterative的算法，比较Messy… 写recursion的效果可能好些…

要是挂估计就挂在这个人身上了…
Onsite 5 印裔
还是上一个问题，怎么进一步优化。（字典用prefix tree来表示，一个Prefix要是到了叶子节点，就没必要继续DFS下去了） coding search on the prefix tree
Onsite 6 老美，也比较Nice
给电话号码，打印可能的words。跟4的第二问差不多，写的recursion，意识到那个题如果写recursion简洁明了的多。
讨论distribute system的设计，考虑durability和availability的balance问题

Atoi OVERFLOW
目标 result*10+str[i]-'0'<=MAX_INT
1) result<MAX_INT/10
2) result=MAX_INT/10 && str[i]-'0'<=MAX_INT/10
接下来就说会问一个分析的问题和一个coding的题目，然后问如何在一个无限长的stream 里面找到前1000大的，因为是非常常见的题目，就和他说了一下用min heap，分析了一下复杂度，很快就进入coding了

13 - 40 min:
问了一道我没有准备过的coding题目，simple regular expression match，可以match 的符号只有3种：

a-z : match a-z
. : match any
* : repeat 0 - arbitrary times

和没用过regex的同学解释一下 例如
a*b 可以match : b, ab, aab, aaaaaaaaaaab
b.*b 可以match : bb, bab, b12345b

我就说我知道regex的实现比较复杂，我没研究过，就只能用比较笨的方法，复杂度应该比较高，他说你先写了看看

函数的prototype是:
public static bool IsMatch(char[] str, char[] pattern)

返回是否match BTW 我是用C#写的，面试的时候觉得比c++好写,有foreach之类的，java也是一样）

##我用的是最笨的方法，扫描pattern，然后递归的match剩下的str和剩下的pattern

由于一开始紧张，上来就写成了wildcard了，，然后面试官说，，你这个不对不是regex，我一看果然，，太紧张了，然后就改了一些（改动不是很大），写的时候那些 error case 都做了判断，其他的也没有什么特别的，写的也不是很好看，后来40分钟快到了，面试官就说一会要开会，留5分钟问他问题，这个时候我的那个基本写完了，但是有点小问题 a*还只能 match 1-n个a，零个的情况没有考虑，然后他说知道了，没关系他知道我的算法了

```
char str[9];
str[8]=’\0’;
printPar(4,4,4,str,0);
```

It is very clear that we will have 4 open and 4 close parentheses. The only logic you have to apply is that right parentheses can appear in output string only when there are already more left parentheses present in the output string. The rest of the comments are in-line with the code.

```
void printPar(int l, int r, char *str, int count) {
if (l < 0 || r < l) return; // invalid state
if (l == 0 && r == 0) {
printf(“%s\n”, str); // found one, so print it
} else {
if (l > 0) { // try a left paren, if there are some available
str[count] = ‘(‘;
printPar(l - 1, r, str, count + 1);
}
if (r > l) { // try a right paren, if there’s a matching left
str[count] = ‘)’;
printPar(l, r - 1, str, count + 1);
}
}
}

struct ListOfLists {
ListOfLists() : next(NULL), data(NULL) {}
ListOfLists* next;
LLNode* data;
};
struct LLNode {
LLNode(Node* t, LLNode* n) : next(n), tree(t) {}
LLNode* next;
Node* tree;
};
ListOfLists* TreeLinkedLists(Node* root) {
ListOfLists * results = new ListOfLists();
traverse(root, results);
return results;
}
void traverse(Node* root, ListOfLists* results) {
if (root == NULL) return; // nothing to do
results->head = new LLNode(root, results->head); // prepend tree node
if (results->next == NULL) {
// extend results if necessary
results->next = new ListOfLists();
}
traverse(root->left, results->next);
traverse(root->right, results->next);
}
```

The main difference between OSPF and RIP is that RIP only keeps track of the closest router for each destination address, while OSPF keeps track of a complete topological database of all connections in the local network. The OSPF algorithm works as described below.
Because of the increase in the population, there is a need of Ipv6 protocol which can provide solution for:

1. Increased address space
2. More efficient routing
3. Reduced management requirement
4. Improved methods to change ISP
5. Better mobility support
6. Multi-homing
7. Security
8. Scoped address: link-local, site-local and global-address space

A mask is a bit pattern used to identify the network/subnet address. The IP address consists of two components: the network address and the host address.
Chess Game:

```
class PositionPotentialValue {
/* compares value of potential game position */
bool operator < (const PositionValue& pv); };
class ChessPieceBase {
virtual void estiationParameter0(); /* used by PositionEstimater
in different circumstances */
virtual int estiationParameter1();
virtual bool canBeChecked();
virtual bool isSupportCastle();
// other rule-base properties
};
class King : public ChessPieceBase { … };
class Queen : public ChessPieceBase { … };
// other chess piece classes
class Position { // represents chess positions in compact form
std::vector<ChessPiece*> black;
std::vector<ChessPiece*> white;
};
class PositionEstimater {
// calculate value of a position
static PositionPotentialValue estimate(const Position& p);
};
class PositionBacktracker {
// get next postion for estimation.
static Position getNext(Position& p);
};
class ChessPieceTurn { ... }; // represents move of a chess piece
class PlayerBase {
virtual ChessPieceTurn getTurn(Position& p);
};
class ComputerPlayer : public PlayerBase {
// actual implementation
void setDifficulty();
PositionEstimater estimater;
PositionBackTracker backtracter;
};
class HumanPlayer : public PlayerBase {...}; // actual implementation
class ChessFormat {...}; // include info about timing, etc.
class GameManager { // keeps track of time, end of the game, etc
void processTurn(PlayerBase * player);
bool acceptTurn(ChessPieceTurn * turn);
Position currentPostion;
ChessFormat formant;
void setGameLog(char * filePath);
};
```

Card Game:

```
class Card {
public:
enum Suit { CLUBS=0x00, SPADES=0x10, HEARTS=0x20, DIAMONDS=0x30 };
Card(Suit s, int r) {
assert(1<=r && r<=13);
card = int(s)+r;
}
virtual ~Card() {}; // just in case
int rank() const { return card & 0x0F; }
Suit suit() const { return Suit(card & 0x30); }
private:
short int card;
};
```

Write a method to randomly generate a set of m integers from an array of size n. Each element must have equal probability of being chosen

```
int[] pickMRandomly(int[] array, int m) {
int[] subset = new int[m];
for(int j = 0; j < m; ++j) {
index = random(j, array.size()-1); // random number in [j,n]
subset[j] = array[index];
array[index] = array[j]; // array[j] is now “dead”
/* potentially unnecessary, depending on how much we can
* modify the array */
array[j] = subset[j];
}
return subset;
}

int rand7() {
while (1) {
int num = 5*(rand5()-1) + rand5() - 1;
if (num < 21) return num % 7;
}
}


/* Returns NULL, p, q, or the nearest common ancestor */
Tree * common_ancestor(Tree * root, Tree * p, Tree * q) {
// assume p and q are in the tree
// will return NULL, p, q, the nearest common ancestor
if (root == null) {
return NULL;
}
if (root == p || root == q) {
return root;
}
left = common_ancestor(root->left, p, q);
right = common_ancestor(root->right, p, q);
/* p is on one side and q is on the other */
if ((left == p && right == q) || (left == q && right == p))
return root;
return left ? left : right;
}

void findsum(node* head, int sum, int[] buffer, int level) {
if (*head == NULL) return;
int temp = sum;
buffer[level] = head->data;
/* Look up through the path we’ve traversed to see if our path
* equals sum */
for (int i = level; i >= 0; i--) {
temp = temp - buffer[i];
if (temp == 0) {
/* We’ve found sum by starting at the current node and
* counting upwards until level i *.
print(buffer, level, i);
}
}
findsum(head->left, sum, buffer, left + 1); // traverse left
findsum(head->right, sum, buffer, left + 1); // traverse right
}
void print(int* buf, int start, int end) {
for(int i = start; i<=end; i++) {
printf(buffer[i]);
}
}
```

FILE SYSTEM:

```
#include <map>
#include <vector>
#include <string>
struct DataBlock {
char data[DATA_BLOCK_SIZE];
};
DataBlock dataBlocks[NUM_DATA_BLOCKS];
struct INode {
std::vector<int> datablocks;
};
struct MetaData {
int size;
Data last_modifed;
Data created;
char extra_attributes;
};
std::vector<bool> dataBlockUsed(NUM_DATA_BLOCKS);
std::map<string, INode *> mapFromName;
sruct FSBase;
struct File : public FSBase {
private:
std::vector<INode> * nodes;
MetaData metaData;
};
struct Directory : pubic FSBase; {
std::vector<FSBase* > content;
};
struct FileSystem {
init();
mount(FileSystem*);
unmount(FileSystem*);
File createFile(cosnt char * name) { /* modifies MetaData */ }
Directory createDirectory (const char * name) { /* modified MetaData */ }
void openFile(File * file, FileMode mode) {
```

```
// mapFromName to find INode corresponding to file
}
void closeFile(File * file) { … }
void writeToFile(File * file, void * data, int num) {
// Modifies all underlying structures.
// Searches for next available datablock with dataBlockUsed
// Modifies INode structure
// Modifies MetaData
// Accesses datablock using INode structure
}
void readFromFile(File * file, void * res, int numbutes, int postion) {
// Accesses datablock using INode structure
};
```

1. 解释Hash Table，包括“可以用什么数据结构实现hash table”，“what is a good hash function”，“什么是load factor”。

2、算法：删除一个给定数列中重复的元素。

3、merge两个有序数组。要求先给他解释算法，再写代码。他当时给了我十分钟的时间，让我写好以后发到他邮箱里。

4、OOD：设计一个汽车出租（Car Rental Agency）的系统。他先问我如果要实现vehicle search，需要哪些类；然后又问要实现rent a car，又需要哪些类；最后问如果快到了交车截至时间，需要向用户发送提醒的邮件，应该怎么做。

第二轮电话面试：
1、Favorite project。

2、你最喜欢的排序算法，它是怎么工作的，有什么优点和缺点。我说的是选择排序，他又问有什么排序算法比它的时间复杂度小，并且同样要描述一下它们是怎么工作的。

3、算法、写代码：
给了两个数组，要求找出他们之中相同的元素，并且将相同的元素存储在一个新的数组里，再输出。如果数组里有重复的元素，比如array1中有四个5，array2中有两个5，那么新数组里只存储两个5。要求我写代码，再念给他听。

4、OOD：设计restaurant reservation系统。

5、好像是一个开放性问题，我没有完全理解：log file中存储了很多order的信息，每个order都有一个oid，字符串格式（xxx-xxxxxxx-xxxxxxx）。要求从log file中把所有的oid都提取出来，问我怎么做。

on-site：
一、给定一篇文章，用户想要搜索一个单词，要求给出搜索单词的建议（就像一般的搜索引擎的那种功能）。要求描述算法、复杂度并写代码。

二、先问了几个行为问题。然后问了两个简单的链表问题，单链表中查找倒数第n个数和判断链表中是否有环。编程题问的是boggle游戏的问题：给定一个4*4的矩阵，每个位置有一个字母，可以从一个位置跳转到周围八个相邻位置中的任何一个，但不能跳到已经访问过的位置，要求找出所有的单词（假设给定了一个词典）。
http://en.wikipedia.org/wiki/Boggle

三、午餐，一个经理问了一些简历上的projects和实习的经历，然后又介绍了一下他们组的工作。他说这次面试我的主要是Kindle组的。

四、不是Kindle组的，我估计是传说中的bar raiser。第一个问题是给了一个很大的文件（不能完全放入内存），其中每一行存一个整数，要求判断这个文件中的数有没有重复。然后是一个开放性问题：一台服务器每过三天就要挂一次，需要重启才能再次使用，每次重启需要一分钟的时间；问有什么方法能解决这个问题。

五、实现一个整数除法的函数，不使用除号，可以使用加号、减号和乘号，函数只返回商

```
#include <vector>
using namespace std;

/* Finds longest strictly increasing subsequence. O(n log k) algorithm. */
void find_lis(vector<int> &a, vector<int> &b)
{
      vector<int> p(a.size());
      int u, v;

      if (a.empty()) return;

      b.push_back(0);

      for (size_t i = 1; i < a.size(); i++)
        {
                // If next element a[i] is greater than last element of
current longest subsequence a[b.back()], just push it at back of "b" and
continue
            if (a[b.back()] < a[i])
                {
                  p[i] = b.back();
                  b.push_back(i);
                  continue;
            }

                // Binary search to find the smallest element referenced by b
which is just bigger than a[i]
                // Note : Binary search is performed on b (and not a). Size
of b is always <=k and hence contributes O(log k) to complexity.
            for (u = 0, v = b.size()-1; u < v;)
                {
                  int c = (u + v) / 2;
                  if (a[b[c]] < a[i]) u=c+1; else v=c;
            }

                // Update b if new value is smaller then previously
referenced value
            if (a[i] < a[b[u]])
                {
                  if (u > 0) p[i] = b[u-1];
                  b[u] = i;
            }
      }

      for (u = b.size(), v = b.back(); u--; v = p[v]) b[u] = v;
}

/* Example of usage: */
#include <cstdio>
```

```
int main()
{
      int a[] = { 1, 9, 3, 8, 11, 4, 5, 6, 4, 19, 7, 1, 7 };
      vector<int> seq(a, a+sizeof(a)/sizeof(a[0])); // seq : Input Vector
      vector<int> lis;                               // lis : Vector
containing indexes of longest subsequence
        find_lis(seq, lis);

        //Printing actual output
      for (size_t i = 0; i < lis.size(); i++)
            printf("%d ", seq[lis[i]]);
        printf("\n");

      return 0;
```

```
bool findMinWindow(const char *str, const char *pattern,
                   int &minWindowBegin, int &minWindowEnd) {
  int N = strlen(str);
  int M = strlen(pattern);
  int minWindowLen = INT_MAX;

  // hash table init all to 0s
  // used to check how many letters left in T to be filled
  char needToFill[256] = {0};

  for (int i = 0; i < M; i++)
    needToFill[pattern[i]]++;

  // set the rest to -1 so we know that letter is not in T
  for (int i = 0; i < 256; i++)
    if (needToFill[i] == 0)
      needToFill[i] = -1;

  // array of queues, each corresponds to a unique char in T
  queue<int> q[256];

  // maintains a sorted map (maps indices to char),
  // the first and last element tells us the
  // starting and ending position of the window
  map<int,char> m;

  for (int i = 0; i < N; i++) {
    // skips characters not in T
    if (needToFill[str[i]] == -1) continue;

    // push character to queue
    if (q[str[i]].size() < needToFill[str[i]]) {
      q[str[i]].push(i);
      m[i] = str[i];
    }
    // replace the character in the queue
    // and updates the corresponding element in the map
    else {
      int idxToErase = q[str[i]].front();
      map<int,char>::iterator it = m.find(idxToErase);
      m.erase(it);
      m[i] = str[i];
      q[str[i]].pop();
      q[str[i]].push(i);
    }

    // found a window, check for minimum
    if (m.size() == M) {
      int end = m.rbegin()->first;
      int begin = m.begin()->first;
      int windowLen = end - begin + 1;
      if (windowLen < minWindowLen) {
        minWindowLen = windowLen;
        minWindowBegin = begin;
        minWindowEnd = end;
      }
    } // end if
  } // end for

  return (m.size() == M) ? true : false;
}
//////////////////////////////////////////////////////////////////////////
bool minWindow(const char* S, const char *T,
               int &minWindowBegin, int &minWindowEnd) {
  int sLen = strlen(S);
  int tLen = strlen(T);
  int needToFind[256] = {0};

  for (int i = 0; i < tLen; i++)
    needToFind[T[i]]++;

  int hasFound[256] = {0};
  int minWindowLen = INT_MAX;
  int count = 0;
  for (int begin = 0, end = 0; end < sLen; end++) {
    // skip characters not in T
    if (needToFind[S[end]] == 0) continue;
    hasFound[S[end]]++;
    if (hasFound[S[end]] <= needToFind[S[end]])
      count++;

    // if window constraint is satisfied
    if (count == tLen) {
      // advance begin index as far right as possible,
      // stop when advancing breaks window constraint.
      while (needToFind[S[begin]] == 0 ||
            hasFound[S[begin]] > needToFind[S[begin]]) {
        if (hasFound[S[begin]] > needToFind[S[begin]])
          hasFound[S[begin]]--;
        begin++;
      }

      // update minWindow if a minimum length is met
      int windowLen = end - begin + 1;
      if (windowLen < minWindowLen) {
        minWindowBegin = begin;
        minWindowEnd = end;
        minWindowLen = windowLen;
      } // end if
    } // end if
  } // end for

  return (count == tLen) ? true : false;
}

int findKthSmallest(int A[], int m, int B[], int n, int k) {
  assert(m >= 0); assert(n >= 0); assert(k > 0); assert(k <= m+n);
  int i = (int)((double)m / (m+n) * (k-1));
  int j = (k-1) - i;

  assert(i >= 0); assert(j >= 0); assert(i <= m); assert(j <= n);
  // invariant: i + j = k-1
  // Note: A[-1] = -INF and A[m] = +INF to maintain invariant
  int Ai_1 = ((i == 0) ? INT_MIN : A[i-1]);
  int Bj_1 = ((j == 0) ? INT_MIN : B[j-1]);
  int Ai   = ((i == m) ? INT_MAX : A[i]);
  int Bj   = ((j == n) ? INT_MAX : B[j]);

  if (Bj_1 < Ai && Ai < Bj)
    return Ai;
  else if (Ai_1 < Bj && Bj < Ai)
    return Bj;

  assert((Ai > Bj && Ai_1 > Bj) ||
         (Ai < Bj && Ai < Bj_1));

  // if none of the cases above, then it is either:
  if (Ai < Bj)
    // exclude Ai and below portion
    // exclude Bj and above portion
    return findKthSmallest(A+i+1, m-i-1, B, j, k-i-1);
  else /* Bj < Ai */
    // exclude Ai and above portion
    // exclude Bj and below portion
    return findKthSmallest(A, i, B+j+1, n-j-1, k-j-1);
}
```

整个算法有点像minimum spanning tree，
z=a(x)+b(y)是个二维函数。我们从原点开始。有把确定的结果涂黑。第一个涂黑的是（1，1）

下一个涂黑的只能是（1，2）或者（2，1），我们只要比较这两个就可以了。假设是（1，2）比较大，那就是（1，2）涂黑。。那下一个candidate只可能在（1，3），（2，1）或者（2，2),

candidate有什么特点呢。就是涂黑的那些点的横坐标或者总坐标加1。

而且可以通过一些前提条件排出一些candidates，比如（2，2）不是candidates，因为（2，1）肯定要比（2，2）大，那candidate其实最多只有min（m，n）个，一行最多只能有一个candidate。所以那些candidates组成一个堆，每次在堆里面删除最大值，也就是把一个点确定涂黑以后，就把它的横坐标或者纵坐标+1的点放入堆。堆的节点数目最多是min（m，n），做k次删除最大值的操作所以复杂度是k log(min(m,n))
写一个loop，从一个文件读数据读到另外一个文件，假定文件都valid，已经open。我写了一个byte by byte的，然后被问了一大堆问题：比如怎么提高效率，buffer size怎么选， 如果line by line的话，line太长了会有什么问题，etc。
对于这一题，最大的bottleneck是disk random seek IO。若是能极大的提高disk sequential seek，那么就可以把random seek的损耗减小到最少。

用byte by byte拷贝最好了，反正buffer size提高点，越高越好，把整个可用的内存都吃掉就没问题。（比如win7操作系统，若是同盘拷贝几百M的文件，内存大于文件体积的话，几乎几秒不到就可以搞定，其实是系统把文件吃到内存里面先，然后在朝目标写的）。

若是有2颗磁盘，那么就用双线程了：一个线程从一颗磁盘里读bytes，一个线程朝另外一个磁盘写bytes，

这样可以
把效率开到最大：读写操作都是sequential seek。

若是line by line的拷贝的话，若是line太长了buffer吃不下，就很麻烦了阿。。最好的办法，是把stream read/write这一层抽象出来，做成BufferedStream。文件拷贝的那一层逻辑就2行代码：

```
while ((line = bufferedStreamIn.readLine()) != null)
    bufferedStreamOut.writeLine(line);
bufferedStreamOut.flush();
```

然后，在bufferedStreamOut里面做实现。给固定的buffersize，每当buffer满的时候，就写入磁盘。

```
void writeLine(String line) {
    foreach char in line:
        if buffer is full
           write buffer into disk
           clear buffer
        end if

        append char into buffer
    end for
}
```

```
boolean hasPath(Point a, Point b) {
   if (a == null || isBlocked(a))
      return false;

   if (a.equals(b))
      return true;

   if (visitedPoints.contains(a))
       return visistedPoints.get(a);

   boolean cango =
           hasPath(goDown(a), b) ||
           hasPath(goRight(a), b) ||
           hasPath(goUp(a), b) ||
           hasPath(goLeft(a), b);

   visitedPoints.put(a, cango);

   return cango;
}
```

动态规划，位运算，递归，图最短路径，堆，杨氏矩阵

Give an algorithm to compress a memory. To be more clear if you are given a memory of some stored data here and there and some empty and null memory in between, how will you fragment and compress your memory?

Design Parking Lot

How do you implement a linked list without using dynamic memory allocation? So basically you need to use an array as a linked list.

Generating and testing Sodoku problem

Given a MxN matrix, in how many ways can you go from top-left to bottom-right?

how can we find the longest palindrome in the given sentence??? Suffix tree

http://www.careercup.com/question?id=3376669

Given two sorted positive integer arrays A(n) and B(n), we define a set S = {(a,b) | a \in A and b
\in B}. Obviously there are n2 elements in S. The value of such a pair is defined as Val(a,b) = a +
b. Now we want to get the n pairs from S with largest values. NlogN

http://www.careercup.com/question?id=3291667

Given a singly linked list sorted in ascending order, convert it to a height balanced BST from this.
http://www.careercup.com/question?id=3327667

Design the data structure to provide the mathematical operations +, - ,/ , * etc for the very very large numbers also implement the + function for two such very very large numbers ...say numbers with 1 Million digits.

{a} There is a string S ans another string s1,
Design the algorithm to check if s1 is contained inside S and return the location as well.
Hint: Interviewer told me that this is a standard problem from book,
{b} test cases to check this.

en.wikipedia.org/wiki/Rabin–Karp_string_search_algorithm
en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm
en.wikipedia.org/wiki/Boyer%E2%80%93Moore_string_search_algorithm

Test cases:
null, null -> exception
null, not null -> exception
not null, null -> exception
empty, empty -> 0
not empty, empty -> -1
empty, not empty -> -1
S smaller than s1 -> -1
S larger than s1 with no occurrences of s1 -> -1
S larger than s1 with 1 occurrence of s1 -> index of s1
S larger, with multiple occurrences of s1 -> first index of s1

if the search is case sensitive, other tests can be imagined.

Implement the function bool isRegex(char *reg, char *string); This function is passed two strings : a regular expression, consisting of the [a-z] and the * and ? characters. We had to check if the string matched the supplied regular expression. For example, if reg is a*b, and string is acbcb, we should return true. And if reg is a?b and string is accb, we return false....

Find closest ancestor of two nodes in a binary tree.

Print a matrix spirally

Reverse an integer array bitwise algorithm? code? Test cases?
http://www.careercup.com/question?id=817692

You are given an array containing only 0,1 and 2. Sort this array in one pass.You can't use anything like counting the no. of 0s and 1s.
http://www.careercup.com/question?id=382535

Given a BST (Binary search Tree) how will you find median in that?
Constraints:
-No extra memory.
-Function should be reentrant (No static, global variables allowed.)
-Median for even no of nodes will be the average of 2 middle elements and for odd no of terms will be middle element only.
-Algorithm should be efficient in terms of complexity.
Write a solid secure code for it.

In usual Intel computers:
Between processes, a context switch needs to change the OS-specific data and VM pointers to data, stack and code segments. But between threads, a context switch would only involves a change to stack segment. The process of swapping and/or signalling switches is handled by the same code (timer interrupt) in the OS.

Write a function to generate all possible n pairs of balanced parentheses.

```
void generate(string p,int o,int c ,int n)
{
if( o==n && c==n )
{
PRINT(p);
return ;
}

if(c>o)return;

if( o >=c && o<n )
generate(p+'{' ,o+1,c,n );

if( c<o )
generate( p+ '}' ,o,c+1,n );

}
```

Connect the leafs in a binary tree and return a reference to the first of the linked leafs. This would allow clients of this API to subsequently traverse the leafs. The API has a fLR flag indicating if the leafs should be connected from left-right (fLR = true) or from right-left (fLR = false).

Please implement the following API:

public Node ConnectLeafs(Node root, bool fLR)

Given a set of points, find the line that intersects the most number of points write a program to shuffle an pack of cards in the most efficient way.

Write code to check if a string contains a substring. KMP, Suffix Tree

given a string, print each character and its number of occurence in sequence. use BST and no recursion, no extra memory is allowed.
e.g, char* str="bcdaceffbe", you should print
a 1 b 2 c 2 d 1 e2 f 2.

http://doc.trolltech.com/qq/qq11-mutex.html

```
QMutex mutex;

    void ReaderThread::run()
    {
        ...
        mutex.lock();
        read_file();
        mutex.unlock();
        ...
    }

    void WriterThread::run()
    {
        ...
        mutex.lock();
        write_file();
        mutex.unlock();
        ...
    }

const int MaxReaders = 32;
    QSemaphore semaphore(MaxReaders);

    void ReaderThread::run()
    {
        ...
        semaphore++;
        read_file();
        semaphore--;
        ...
    }

    void WriterThread::run()
    {
        ...
        semaphore += MaxReaders;
        write_file();
        semaphore -= MaxReaders;
        ...
    }
```

```
    10
6 12

4 -> 5 <-------->11<-----> 13 <--
```

|_________________________________|
In the BST you have the leaf nodes connected to form a doubly LL. Given a node, identify its height

In our indexes, we have millions of URLs each of which has a link to some page contents, that is, URL->contents. Now, suppose a user types a query with wild cards *, which represent 0 or multiple occurrences of any characters, how do you build the indexes such that such a type of query can be executed efficiently by finding all corresponding URLs->contents efficiently. For example, given a query http://www.*o*ve*ou.com. You need to find iloveyou.com, itveabcu.com, etc

given two binary search trees, merge them in O(n) time with O(1) space

http://www.careercup.com/question?id=254667

Given char* func1(char* target, char* substring,char* replacement)
write a c++ code to find the substring in the target and replace the whole substring with the replacement. (hint: replacement can be larger or smaller than the substring.)consider all possible test cases and check.

merge two sorted list;
3. rotate an array by K;
4. Given an XML string, check if it's valid or not;
5. binary tree: each node has an additional field node which is initialized to be NULL at first. Asked to, for each node, point its next pointer to the next node in level-by-level traversal order. NO QUEUE should be used HERE!

Given two arrays of numbers, find if each of the two arrays have the same set of integers ? Suggest an algo which can run faster than NlogN without extra space?

The requirement is to get the size of the datatype, without declaring a variable or a pointer variable of that type,And, of course without using sizeof operator !

To find LCA for nodes A and B:

O((logn)^2):
1. Find in A in left subtree, B in right subtree
2. If both not found, find in A in right subtree, B in left subtree
3. If both found, current node is the common LCA
4. If one found and not the other, make a recursive to call to that branch of the tree and start from 1.

O(nlogn) with O(n) space:
1. Traverse the tree until node A is found, store the path in an array a1.
2. Traverse the tree until node B is found, store the path in an array a2.
3. Compare a1 and a2, the last common element is the LCA.

But there are better, more complicated ways of doing this in constant time using RMQ.

http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=lowestCommonAncestor#Lowest%20Common%20Ancestor%20%28LCA%29

find the longest palindrome in a string?

Given a character string, find out all the English words contained in the string. Optimize the solution.

Given a M*N matrix A in which all the elements in a row and all the elements in a column are strictly increasing. Find a path from the smallest element (ie A[0][0]) to the largest element (ie A[M-1][N-1]) such that the sum of the elements in the path is maximum. Time Complexity O(m+n). Use efficient space.   Topological sort.

Peterson's algorithm is a concurrent programming algorithm for mutual exclusion that allows two processes to share a single-use resource without conflict, using only shared memory for communication. It can be extended to more than two processes. Using Peterson's algorithm to implement mutual exclusive access to stack:

```
while(1) { // thread i (0 <= i < n)
for (j=1 ; j<n ; j++) {
flag[i]=j;
last[j]=i;
for (k=0 ; k<n ; k++) {
if (k==i) continue;
while (flag[k]>=flag[i] && last[j]==i) {
sleep(random());
}
}
}
// critical section
…
// end of critical section
flag[i]=0;
// not critical section
…
// end of not critical section
}
```

http://en.wikipedia.org/wiki/Suffix_tree

min/max – heap

Maximum subarray sum problem

Kadane's 2D algorithm O(N^3)

```
#include <iostream>
#include <algorithm>

using namespace std;

int main( void )
{
    int N;
    int t = 0;
    int a[100][100];
    int pr[100];
    int S = 1<<31, s = 0, k, l, x1 = 0,x2 = 0,y1 = 0,y2 = 0,j;

    cin >> N;

    for( int i = 0; i < N; i++)
        for( j = 0; j < N; j++)
            cin >> a[i][j];

    for( int z = 0; z < N; z++)
    {
        for(int i = 0; i < N; i++) pr[i] = 0;
        for(int x = z; x < N; x++)
        {
            t = 0;
            s = 1<<31;
            j = 0;
            k = 0; l = 0;
            for(int i = 0; i < N; i++)
            {
                pr[i] = pr[i] + a[x][i];
                t = t + pr[i];
                if( t > s)
                {
                    s = t;
                    k = i;
                    l = j;
                }
                if( t < 0 )
                {
                    t = 0;
                    j = i + 1;
                }
            }
            if( s > S)
            {
                S = s;
                x1 = x;
                y1 = k;
                x2 = z;
                y2 = l;
            }
        }
    }

    cout << x1 << " " << y1 << " " << x2 << " "  << y2 << endl;
    cout << S;

    return 0;
}
```

Given an array, find the longest subarray which the sum of the subarray less or equal then the given MaxSum
int[] FindMaxSumArray(int[] array, int maxsum)

http://www.careercup.com/question?id=209667

Given a string of 10 characters and a number, insert multiplies and
additions to make the characters equal the number
ie
1232537859, 995 -> 123*2+35*7+8*58

Given an array of N elements , one element is repeated N/2 times. Find the element if such an element exists.

You have a stack that is accessed by multiple threads simultaneously and you wish to synchronize access. You do not want to use locking to implement synchronization. Implement a thread-safe version of the stack.

http://www.careercup.com/question?id=84766

Serialize:

```
#include<vector>
#include<string>
#include<iostream>
using namespace std;



int serialize(const vector<string> & stringVector1)
{
        FILE *fptr = fopen("C:\\serializedString.txt","w");
        if(!fptr)
                return 0;



vector<string>::const_iterator i = stringVector1.begin(), end =
stringVector1.end();
        for(; i<end;i++ )
        {
                ::fputs((*i).c_str(),fptr);
                ::fputs("\n",fptr);
        }
        fclose(fptr);
        return 0;
}
int Deserialize(vector<string> & stringVector2)
{
        FILE *fptr = fopen("C:\\serializedString.txt","r");
        if(!fptr)
                return 0;



char arr[1024],*ptr=NULL;
        while(!feof(fptr))
        {
                ptr = fgets(arr,1024,fptr);
                if(ptr == arr)
                        stringVector2.push_back(string(arr));
        }

        fclose(fptr);
        return 0;
}
```

```
int main()
{
	vector<string> stringVector1,stringVector2;
	stringVector1.push_back("Cracking");
	stringVector1.push_back("Programming");
	stringVector1.push_back("Interview");
	stringVector1.push_back("with");
	stringVector1.push_back("Arif");
	stringVector1.push_back("and");
	stringVector1.push_back("Krishna");
	serialize(stringVector1);
	Deserialize(stringVector2);
	vector<string>::iterator i = stringVector2.begin(), end = 
stringVector2.end();
	for(; i<end;i++ )
		cout<<*i;
	getchar();
	return 0;
}
```

implement your own malloc and free for application x, which should control the heap memory usage of the application x.

Given an input array of integers of size n, and a query array of integers of size k, find the smallest window of input array that contains all the elements of query array and also in the same order.

Create methods for Set implementation. (Getting unique values from user to create a Set, and methods to implement Intersection, Union... of 2 sets

you are given 2 arrays sorted in decreasing order of size m and n respectively.
Input: a number k <= n*n and >= 1
Output: the kth largest sum(a+b) possible. where
a (any element from array 1)
b (any element from array 2)

Given preorder and inorder traversal of tree, write the code to form binary tree from given traversal.

```
Tree* Createnode(int *inorder, int *preorder, int index, int high, int low, int length)
{
int element = preorder[index];
int pos = search(element, inorder, low, high);
if(pos >= 0)
{
Tree *temp = new Tree();

temp->data = element;
temp->left = Createnode(inorder, preorder, index+1, pos-1, low, length);
temp->right = Createnode(inorder, preorder, index+1, high, pos+1, length);

return temp;
}
else
{
if(index < length && low <= high)
return Createnode(inorder, preorder, index+1, high, low, length);
else
return NULL;
}

}

int _tmain(int argc, _TCHAR* argv[])
{
int inorder[] = {11,3,6,13,9,15,8,14};
int preorder[] = {6,3,11,8,9,13,15,14};

Tree *root = NULL;

root = Createnode(inorder, preorder, 0, 7, 0, 8);
return 0;
}
```

Interval trees

Write a function to validate a SuDoKu board.

Write a function void DrawRectangle(char *Screen, int x1, int y1, int x2, int y2). Height and width of the monitor is known. To set a pixel, you need to set that particular bit of the screen.

```
void DrawRectangle(char *Screen, int x1, int y1, int x2, int y2)
{
if (x1>=width || x2>=width || x1>x2 || y1>=height || y2>=height || y1>y2)
return;
for (int i=y1; i<=y2; i++)
{
if (i==y1 || i==y2)
{
for (int j=x1; j<=x2; j++)
setPixel(j,i);
}
else
{
setPixel(x1,i);
setPixel(x2,i);
}
}

}

void setPixel(char *Screen, int x, int y)
{
int n = (y>0) ? y*width : 0;
n+=x;
int a = n/8;
int b = n%8;
Screen[a]+= 256>>b;
}
```

An array of integers of size n-1, all the elements are form [1,n]. Find the missing number. You can read only one bit in one operation, ie, to read A[i], you need to perform log(A[i]) operations.

Given two sorted postive integer arrays A(n) and B(n) (W.L.O.G, let's
say they are decreasingly sorted), we define a set S = {(a,b) | a \in
A and b \in B}. Obviously there are n^2 elements in S. The value of such
a pair is defined as Val(a,b) = a + b. Now we want to get the n pairs
from S with largest values.

Auto pointers have ownership. If you assing an auto pointer to another, the assigned auto_ptr loses ownership. Point to note is that RHS is modified.

Shared pointers are reference counted. Assignment or copying increases the count and out of scope or delete reduces the count. When the count goes to 0, actual object is destroyed. All you go to do in a simple case is to maintain a counter which is incremented in constructors and decremented in destructors. If count goes to 0, delete the actual object. If a robust solution is needed Boost::shared_ptr is available.

Given an 32-bit integer X, swap the i-th and j-th bit.

```
swap(int n,int i,int j)
{
if( (n & 1<<i)>>i ^ (n & (i<<j))>>j) ) // if bits i and j are different
{
n^= 1<<i;
n^= 1<<j;
}
return n;
}
```

Describe and algorithm and implement UNIX tail command

```
void tail(FILE *fp, int num_lines)
{
char **buffer = calloc(num_lines, sizeof(char *));
int index = 0;
int i;

for (;;)
{
char *line = NULL;
int len = 0;

if (getline(&line, &len, fp) < 0)
break;

if (buffer[index])
free(buffer[index]);

buffer[index] = line;

index++;
index %= num_lines;
}

for (i = 0; i < num_lines; i++)
puts(buffer[(index + i) % num_lines]);
}
```

give an array, remove all the 'a's and add one 'd' after each 'b', do it in O(n)...

```
int i=0, pos = 0, nb = 0, ;
for(int i=0; i<n; i++) {
    if(a[i] == 'b') {
          nb++;
    }
    if(a[i] != 'a') {
        a[pos] = a[i];
        pos++;
    }
}
if(nb==0) return;
i = pos;
pos = pos+nb;
for(; i>=0; pos--,i--) {
    if(a[i]=='b') {
        a[pos] = 'd';
        pos--;
        a[pos] = 'b';
    } else {
        a[pos] = a[i];
    }
}
```

In an ordinary sorted list, `insert`, `remove`, and `find` operations require sequential traversal of the list. This results in $O(n)$ performance per operation. Skip Lists allow intermediate nodes in the list to be ``skipped" during a traversal - resulting in an *expected* performance of $O(\lg n)$ per operation

http://www.csee.umbc.edu/courses/undergraduate/341/fall01/Lectures/SkipLists/skip_lists/skip_lists.html

There is a stream of numbers, design an effective datastructre to store the numbers and to return the median at any point of time.

Maintain 2 heaps : min and max
Each time when a number is to be inserted. Insert it in max first. Now get the number at the top and insert it in min heap. The number at the root in the min heap will give you the median.
[or] half of the top elements of max and min heap, if n is even.

[Note: Each time keep equal no. of element in min and max heap. Also adjust if the max element in max heap is larger than the min element in min heap. Exchange the two numbers and rebuild the min heap]

Adding each number will take O(log n) time

Write an algorithm to check the winning condition in a tic-tac toe game for a NXN grid ? (Hint . can be done in O(1) need int ROW[N]; int COL[N]; int diagonal; int anti-diagonal)

```
#define SIZE 3
#define MSIZE -3
```

```
int row[SIZE];
int col[SIZE];

int init () {
int i, j;
for (i=0; i<SIZE; ++i)
row[i] = 0;
for (j=0; j<SIZE; ++j)
row[j] = 0;
}

// 0: no win
// 1: 1st player win
// 2: 2nd player win
int check (int x, int y, int p){ // O(1)
if (p == 1){
row[y]++;
col[x]++;
}
else {
col[x]--;
row[y]--;
}

if (row[y] == SIZE ||
row[y] == MSIZE ||
col[x] == SIZE ||
col[x] == MSIZE)
return p; //

return 0;
}
```

Given an array of 'n' random integers. Given an integer k<=n. Find the k numbers such that the minimum difference of all the possible pairs of k numbers is maximum (maximum among other minimum differences for various possible selections of k numbers).

How to increase web browsing speed. You are allowed to do anything at client/server

At the server:
Enable gZip.
Ensure caching of static file fetch is enabled.
If your users are spread across the globe, try having servers located in diff GeoLocs to save the n/w latency. Content Management System might also can take adv of it.
router/firewall can also do some caching. see, if you can take advantage of that.
On the client:
Limit roundtrips to minimal.
Keep javascript outside of the page, into one(or more) js files, so that they can be used in multiple pages, with taking advantage of caching.
club css files into a single(or as minimum nos as possble) file.
use image-crunching for tiny/small images used in the web pages
make use of ajax, to avoid full-page round-trips.

how to implement a queue using one integer. this should store value 0 to 9. example suppose queue has first value 2 then insert 4 then 6 so it should look like 246. first value should be popped as 2. then it should be 46. program should support 0 in all the levels also. example queue should handle like 01235 also, 0 as first value in queue. remember 0 just to use integer, nothing else as data storage.

Given an array of integers(both positive and negative) divide the array into two parts(sub-arrays) such that the difference between the sum of elements in each array is minimum????

n an array of n elements, find if there is a subset of 3 elements sum up to value T with time complexity O(nlgn).

Design an LRU cache with all the operations including getting the least recently used item to be O(1).

Use a Hashmap along with doubly linked list.
Insertion: When an element is inserted into the hashmap create a new node at the front of the doubly linked list. The hashmap entry will have the reference to this node in the douly linked list. Also the node in the liked list will have a reference to the entry in the hashmap.
So Insertion : O(1)
Deletion: Delete the entry from the hashmap and also following the reference to the doubly linked list, delete the node too.
So O(1)
Access: Get the element in the hashmap, follow the reference to the doubly linked list and just move this node in the doubly linked list to the front of the list.
Recently used: Get the first element from the linked list.
So Access: O(1)

Largest rectangle in histogram problem

http://homeofcox-cs.blogspot.com/2010/04/max-rectangle-in-histogram-problem.html

```
#include <iostream>
#include <sstream>
#include <stack>
using namespace std;

int DEBUG = 0;


void getMax(int hist[], stack<int> * s,
            int newHeight, int right, int & max, int & start) {
  int height, left = 0, area;
  while (s->size() > 0 && hist[s->top()] > newHeight) {
    height = hist[s->top()];
    s->pop();
    left = (s->size() > 0) ? s->top() : start;
    while (s->size() > 0 && hist[s->top()] == height) {
      s->pop();
      left = (s->size() > 0) ? s->top() : start;
    }

    area = height * (right - left);
    if (DEBUG) printf("area: %d * (%d - %d) = %d\n", height, right, left, area);
    if (area > max) max = area;
  }
}

//
// Note that when hist[i] == top_v, we push i.
// In the acm judge site, it says skip i if equal.
// But I feel somehow it can't keep track of the left value
// when multiple columns have the same height.
//
int doHist(int hist[], int len) {
  stack<int> * s = new stack<int>;
  int i, max, top_v;
  int start = -1; // the position before the last 0, used by left.

  max = 0;
  for (i = 0; i < len; i ++) {
    if (s->size() == 0) {
      s->push(i);
      continue;
    }

    top_v = hist[s->top()];
    if (hist[i] >= top_v) {
      s->push(i);
    } else if (hist[i] < top_v) {
      getMax(hist, s, hist[i], i - 1, max, start);
      s->push(i);
      if (hist[i] == 0) start = i - 1;
    }
  }

  getMax(hist, s, 0, i - 1 , max, start);

  cout << "max = " << max << endl;
  return max;
}

int main() {
  int hist[] = {3, 5, 4, 7, 6, 5, 2}; // answer: 20
  doHist(hist, sizeof(hist) / sizeof(int));
  return 0;
}
```

How would you find the first unique url among the millions of url available?

BFS, DFS, Detecting Cycles,

Convert a doubly linked list to a binary search tree in place.

Given a 2D plane, suppose that there are around 6000 points on it. Find a line which passes the most number of points.

unclockwise BST:

```
void PrintLeafNode(Tree *root)
{
if(root != NULL)
{
if(root->left == NULL && root->right == NULL)
printf("\n %d", root->data);
PrintLeafNode(root->left);
PrintLeafNode(root->right);
}
}

void PrintLEdges(Tree *root)
{
if(root != NULL)
{
printf("\n %d", root->data);
PrintLEdges(root->left);
PrintLeafNode(root->right);
}
}

void PrintREdges(Tree *root)
{
if(root != NULL)
{
PrintLeafNode(root->left);
PrintREdges(root->right);
printf("\n %d", root->data);
}
}

int _tmain(int argc, _TCHAR* argv[])
{
int arr[] = {50,40,70,30,45,60,90,42,47,55,65,80};
Tree *root = NULL;

for(int i=0; i<11; i++)
root = CreateTree(root, arr[i]);

printf("\n %d",root->data);
PrintLEdges(root->left);
PrintREdges(root->right);
return 0;
}
```

clockwise:

```
void FlipOrder(Node* root)
{
if (root == null) return;
if( root->left)
{
Print(Node->left->data);
FlipOrder(Node->left;
}
if( root->right)
{
FlipOrder(Node->right->data);
Print(Node->right->data);
}
Print(Node->data);
}
```

```
int MyAtoi(char *inputParam){
  if(*inputParam == 0) throw new FormatException();
  int result = 0;
  bool isNegative = false;
  if(*inputParam == '-'){
      inputParam++;
      isNegative = true;
  }
  if(*inputParam == 0)  throw new FormatException();
  while(*inputParam){
     if(*inputParam >= 0x32 && *inputParam < 0x3C){
        int tempResult = (result
                    + (*inputParam - 0x32)
                 )
                 * ((*(inputParam + 1) == 0) ? 1 : 10);
        if(tempResult < result) throw new OverflowException();
        result = tempResult;
     } else {
       throw new FormatException();
     }
  }
return result * (isNegative ? -1:1);
}
```

Reverse a singly linked list

```
	//
	// iterative version
	//
	Node* ReverseList( Node ** List )
	{

		Node *temp1 = *List;
		Node * temp2 = NULL;
		Node * temp3 = NULL;

		while ( temp1 )
		{
			*List = temp1; //set the head to last node
			temp2= temp1->pNext; // save the next ptr in temp2
			temp1->pNext = temp3; // change next to privous
			temp3 = temp1;
			temp1 = temp2;
		}

		return *List;
	}
```

```
/***********************************************************************

-> This C++ Program is to convert a given infix expression
   (either parenthesized or unparenthesized) to postfix form

-> Ex. of infix expression is ::(a+b^c^d)*(c+d)

-> Data Structers used
     Stack:array

-> This program works in microsoft vc++ 6.0 environment.

*************************************************************************
/

#include<iostream.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>

class expression
{
private:
 char infix[100];
 char stack[200];
 int top;
 int r;
 char postfix[100];
public:
 void convert();
 int input_p(char);
 int stack_p(char);
 int rank(char);
};
int expression::input_p(char c)
{
 if(c==’+’ || c==’-')
  return 1;
 else if(c==’*’ || c==’/')
  return 3;
 else if(c==’^')
  return 6;
 else if(isalpha(c)!=0)
  return 7;
 else if(c==’(‘)
  return 9;
 else if(c==’)')
  return 0;
 else
 {
  cout<<”Invalid expression ::input error\n”;
  exit(0);
 }
}
int expression::stack_p(char c)
{
 if(c==’+’ || c==’-')
  return 2;
 else if(c==’*’ || c==’/')
  return 4;
 else if(c==’^')
  return 5;
 else if(isalpha(c)!=0)
  return 8;
 else if(c==’(‘)
  return 0;
 else
 {
  cout<<”Invalid expression  ::stack error\n”;
  exit(0);
 }
}
int expression::rank(char c)
{
 if(c==’+’ || c==’-')
  return -1;
 else if(c==’*’ || c==’/')
  return -1;
 else if(c==’^')
  return -1;
 else if(isalpha(c)!=0)
  return 1;
 else
 {
  cout<<”Invalid expression ::in rank\n”;
  exit(0);
 }
}
void expression::convert()
{
  cout<<”\n**************************************************\n”
  <<”This program converts the given infix expression\n”
  <<”in to postfix form”
                <<”\n**************************************************\n”;
 cout<<”Enter an infix expression ::\n”;
 cin>>infix;
 int l=strlen(infix);
 infix[l]=’)';
 infix[l+1]=”;
 //Convertion starts
 top=1;
 stack[top]=’(‘;
 r=0;
 int x=-1;
 int i=0;
 char next=infix[i];
 while(next!=”)
 {
  //Pop all the elements to outputin stack which have higher precedence
  while( input_p(next) < stack_p(stack[top]) )
  {
   if(top<1)
   {
    cout<<”invalid expression ::stack error\n”;
    exit(0);
   }
   postfix[++x]=stack[top];
   top–;
   r=r+rank(postfix[x]);

   if(r<1)
   {
    cout<<”Invalid expression  ::r<1\n”;
    exit(0);
   }
  }
  if(input_p( next ) != stack_p( stack[top]))
   stack[++top]=next;
  else
   top–;
  i++;
  next=infix[i];
 }
 postfix[++x]=”;
 if(r!=1 || top!=0)
 {
  cout<<”Invalid expression ::error in rank or stack\n”;
  exit(0);
 }
 cout<<”\n\nThe corresponding postfix expression is ::\n”;
 cout<<postfix<<endl;
}
int main()
{
 expression obj;
 obj.convert();
 return 0;
}
/**********************************************************************
*
SAMPLE OUTPUT::
——————

************************************************
This program converts the given infix expression
in to postfix form
************************************************
Enter an infix expression ::
(a+b^c^d)*(c+d)
The corresponding postfix expression is ::
abcd^^+cd+*
Press any key to continue
**********************************************************************
**/
```

Microsoft

Interview 1 (phone interview)

1. What's your most challenging problem in your projects? (I try to use the STAR principle: situation, task, action, result.)
2. If you can go back in time, what would you do better?
3. What's your preference for the positions in MS? (testing or developer)

4. What's your experience in testing?
5. What's a binary search tree?
6. Suppose you are going to explain this concept to a 5 year old girl, how are you going to explain it?
7. How to test a calculator (mouse/chair/glasses/whatever)?
8. How to get an applicant's telephone number if you know: First name, last name, school, email address? (Pressure question, he will push you for more answers. Prepare for at least 10 solutions)
9. What's the use of binary search tree?

Onsite Interview (SDET in live team)

Summary of questions:

1. Sort 0,1 bit array. Use partition and count.
2. let me give test cases for one of her GUI applications.
3. Given a movie star relation (co-star in one movie) database, given a most popular star (say A), then find the distance of other star to A. BFS.
4. How to convert an integer array to byte array? How to test elevator?

```
byte MyBytes[4];  //set values to this also.
int Int32 = 0;

Int32 = (Int32 << 8) + MyBytes[3];
Int32 = (Int32 << 8) + MyBytes[2];
Int32 = (Int32 << 8) + MyBytes[1];
Int32 = (Int32 << 8) + MyBytes[0];
```

```
uint32 GetInt32( uint8 *pBytes )
{
return (uint32)(*(pBytes + 3) << 24 | *(pBytes + 2) << 16 | *(pBytes + 1) << 8 | *pBytes);
}

void Int32ToUInt8Arr( int32 val, uint8 *pBytes )
{
pBytes[0] = (uint8)val;
pBytes[1] = (uint8)(val >> 8);
pBytes[2] = (uint8)(val >> 16);
pBytes[3] = (uint8)(val >> 24);
}
```

5. How do you feel about today’s interview, how much things do you learn.

Google

Interview 1

1. Implement a code to do wildcast string matching.

```
int main()
{
    char* text = "scr*w?d";
    char* sig = "screeeewywxd";
    int i = 0;
    int j = 0;
    for(i = j = 0; text[i] && sig[j]; ++i )
    {
        if( text[i] == '*')
        {
            while( sig[j] && (sig[j] != text[i+1]) )
                j++;
        }
        else if( sig[j] == text[i] || text[i] == '?' )
        {
            j++;
        }
        else
        {
            printf("Match failed\n");
            break;
        }
    }

}
```

e.g. source: readme.txt, query: *.txt, should return true.

2. check whether a Sudoku is valid. 9*9 matrix, and each row, column and 3*3 cell only contain unique integers (in range [1,9]) or empty.

```
bool isValid(int grid[] [9])
{
        int i, j;
        bool status;
        status = true;

        for (int column = 0; column < 9; column++)
                if (column != j && grid[i] [column] == grid[i] [j])
                        status = false;
cout << "1st test: " << status << endl;

        for (int row = 0; row < 9; row++)
                if (row != i && grid[row] [j] == grid[i] [j])
                        status = false;
cout << "2nd test: " << status << endl;

        for (int row = (i / 3) * 3; row < (i / 3) * 3 + 3; row++)
                for (int col = (j / 3) * 3; col < (j / 3) * 3 + 3; col++)
                        if (row != i && col != j && grid[row] [col] == grid[i] [j])
                                status = false;
cout << "3rd test: " << status << endl;

        for (int i = 0; i < 9; i++)
                for (int j = 0; j < 9; j++)
                        if (grid[i][j] != 0)
                                status = false;
cout << "4th test: " << status << endl;

      for (int i = 0; i < 9; i++)
            for (int j = 0; j < 9; j++)
                  if ((grid[i][j] < 0) || (grid[i][j] > 9))
                        status = false;
cout << "5th test: " << status << endl;


      return status;

}
```

3. Find intersection of two sorted array A, B.

```
vector<int> findIntersection(vector<int> A, vector<int> B) {
  vector<int> intersection;
  int n1 = A.size();
  int n2 = B.size();
  int i = 0, j = 0;
  while (i < n1 && j < n2) {
      if (A[i] > B[j]) {
          j++;
      } else if (B[j] > A[i]) {
          i++;
      } else {
          intersection.push_back(A[i]);
          i++;
          j++;
      }
  }
  return intersection;
}
```

i) What if elements of array B is stored on disk, and the memory is limited such that you cannot load all elements into the memory at once?
ii) How will the complexity change in this case? Are there any factors you need to consider?
iii) How do you change your solution to adapt to this situation?

All above questions need to write detailed codes, check input, and handle special cases. Need to provide time/space complexity.

Interview 2
1. Lots of compiler stuff which I know nothing.
2. Check whether a binary tree is a binary search tree.

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>

/* A binary tree node has data, pointer to left child
   and a pointer to right child */
struct node
{
    int data;
    struct node* left;
    struct node* right;
```

```
};

int isBSTUtil(struct node* node, int min, int max);

/* Returns true if the given tree is a binary search tree
 (efficient version). */
int isBST(struct node* node)
{
  return(isBSTUtil(node, INT_MIN, INT_MAX));
}

/* Returns true if the given tree is a BST and its
   values are >= min and <= max. */
int isBSTUtil(struct node* node, int min, int max)
{

  /* an empty tree is BST */
  if (node==NULL)
     return 1;

  /* false if this node violates the min/max constraint */
  if (node->data < min || node->data > max)
     return 0;

  /* otherwise check the subtrees recursively,
   tightening the min or max constraint */
  return
    isBSTUtil(node->left, min, node->data) &&
    isBSTUtil(node->right, node->data+1, max);
}

/* Helper function that allocates a new node with the
   given data and NULL left and right pointers. */
struct node* newNode(int data)
{
  struct node* node = (struct node*)
                       malloc(sizeof(struct node));
  node->data = data;
  node->left = NULL;
  node->right = NULL;

  return(node);
}

/* Driver program to test above functions*/
int main()
{
  struct node *root = newNode(4);
  root->left        = newNode(2);
  root->right       = newNode(5);
  root->left->left  = newNode(1);
  root->left->right = newNode(3);

  if(isBST(root))
    printf("Is BST");
  else
    printf("Not a BST");
  getchar();
  return 0;
}
```

Need to write detailed codes, time/space complexity, any improvements?
3. Sampling of incoming integers, then return one sample with equal probability.
Time/space complexity, how to prove you are right?

Interview 3
(Host bidding 1)
1. Ask general description of my related projects.
2. Give a general description of his potential project.
3. discuss about some implementation details.

Interview 4
(Host bidding 2)
He has really exciting project and match my background perfectly, many technical questions though, unexpected. The lesson here is that expecting technical questions even in host bidding interviews.
1. Describe in detail of your previous related project. (Android, Google API, PhD research)
2. The major advantages and disadvantages of following languages: C++, Python, Java. (He asked for at least 3 disadvantages for each language, if you can only give two, he will continue to let you think).
3. What's the difference between C# and Java, why you choose C# in one of your project?

Feature	C#	C++	Java
Inheritance	Single class inheritance, multiple interface implementation	Multiple class inheritance	Single class inheritance, multiple interface implementation
The notion of interface	Through the "interface" keyword	Through abstract class	Through the "interface" keyword
Memory management	Managed, using a garbage collector	Manual	Managed, using a garbage collector
Pointers	Yes, but only in the rarely-used unsafe mode. References are used, instead.	Yes, a very commonly used feature.	Not at all. References are used, instead.
Form of Compiled Source Code	.NET intermediate language (IL)	Executables.	Byte code.
One common base class	Yes	No	Yes

4. Consider you are constructing a system for data synchronization, what problem will you face, and how you solve it? (I did not do well on this question, since for my understanding, the data synchronization is normally among process, or among different users, like the one in source code version control (Git/repo). I finally understand after 15 mins, he wants to know about multi-threads synchronization. :<)

5. What is mutex, semaphore, deadlock? Give examples of them. (That's when I finally realize what he wants to know about synchronization, just the classic stuff.)

Interview 5 (Host bidding 3)
The interview runs very smoothly. He basically just talked about which experiences I have.

Facebook:
First phone screen.
1. Tell me about your self, your PhD research, what do you want to do in facebook.
2. What's your applications/projects in Garmin?
3. Do you use facebook a lot? What do you normally do in using it.
4. Binary search, complexity.
5. Bubble sort, best case complexity.
6. Guess a number in a given range, say 1000. (still Binary search).
7. When will java destruct object. (automatic garbage collection for unused object that no reference points to it, finalize() method)
8. Java stuff: how to avoid other programmer from changing the function. (Final keyword)
9. What is the transient keyword.

Second Phone Interview.
1. Describe your background, and what you are seeking for. Then he tell me I am not a good fit for his team, and want to recommend me to the other team. He even didn't want to continue the interview. :<
2. How to use stacks to simulate queue. (do not use online tool, just write and tell him. Use two stacks).
3. How to find the lowest common ancestor of a binary tree, node do NOT have parent pointers. (Recursion, additional check for the case when nodes are not in the tree, or only one node is in the tree.) Use collabedit.com, really awesome tool.

Third Phone Interview.
1. The project they are working on.
2. The projects I was working in research and internship, all resume stuff.
3. Given a linked list, say A->B->C, print it in reversed order. Time & space analysis. What if I want the original list not changed? How about multithreads call this functions simultaneously?

第一个人
1. 返回给定字符串的第一个不符合字母顺序的index
比如abcdaf就需要返回第二个a的index，比如aZe就返回e的index
反正题目不难，就是需要考虑的东西多一点

2. 检查sudoku的输入是valid，允许solution是不完全的
题目一样还是简单，还是要考虑一些细节
比如你的matrix用什么表示，int**的话 就没法表示空白的格子

第二个人
1. 就是那道带random pointer的linklist的copy。我当时给的solution当然是用hashmap之类的东西做，她也没反对，好像是接受了。不过刚才看了其他的解，觉得最好的还是那个在原有序列里面插入再拆分的解，看着比较完美

2. 不是程序问题了，稍微有点设计的意思。给了3个函数，分别是注册一个电话号码；是否被注册；返回一个未被使用的号码。然后问我数据结构以及方法怎么实现。我首先问这些号码是存储在内存还是数据库里面的。其实这个题目挺二的，哪有程序会把号码存内存里面的，不过我猜她想考察这个东西，所以故意这么问的。这个题目答的不是太好，首先是需求没有搞清楚，就是电话号码是否全在里面了。是否允许创建一个新号码。她自己也没弄明白，说你可以随意产生一个号码。这样一来就把题目弄的很麻烦。其实本意显然是所有号码都在里面了。然后就是数据结构了，当时有点紧张，想到的当然是hash，后来因为发现如果用一个hash的话，返回一个没使用过的号码很费时间（假如号码库很大），于是就又拆成了两个存储，一个是使用中的，一个是没有被使用的。也不知道有没有更好的数据结构了

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
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 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
- Bubble/select/insertion/counting/qsort/heap/merge/bst
- Time/space complexity analysis

- Caching design
  - Replacement policy (LRU, LFU, NRU, etc…)
  - Efficiency/complexity/performance
  - Distributed cache
  - Hashing

- Multi-thread
  - Locking/mutex/semaphore/critical section
  - Coding pattern
    - producer/consumer (aka writer/reader)
    - ref-counting

- Tree
  - Heap
  - BST/black-red (BR/AVL is a stretch)
  - B+
  - Suffice/prefix trees (trie)
  - Expression tree

- DP (dynamic programming)

- Search
  - Basic backtrack
  - Backtrack trimming
  - Breadth first vs depth first search
  - A*
  - Bidirectional search

- Graph
  - Shortest distance (djstra, floydwarshall)
  - Cycle detection
  - Flow network algorithms

- List
  - Stack/queue/priority queue
  - List manipulation

- Pattern matching
  - Strict string matching
  - Wildcard matching

- Bit operations

- Compression algorithm.

- Huffman
- RLE (run-length encoding)
- LZW

- Design
  - Design pattern: singleton, factory, provider, witness, command, etc
  - OOP designs
    - Interface vs abstract class
    - Virtual behaviors
  - Design a feature like:
    - Twitter
    - Facebook/linkedin friend recommendation

- C#/Java
  - GC algorithm

- Tech
  - Hadoop
  - Mapreduce
  - TSQL/SQL
  - Memcached
  - Membase
  - 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 ?

3.

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.

1. 我们知道，从一个数组里找一段(连续的)子数组求最大和，是一道经典的面试题，方法很简单，只要O(n)的时间。把这个问题变一下，假设是一个循环数组呢？找一个size<=n的子数组with最大和。

   分析，很容易想到第一步，找个地方把循环数组切断，回到了原来的问题，然后在考虑一下额外的情况。额外的情况就是：有可能最大和的子数组是跨越了切断点的？这种情况的最大和怎么求呢？一个naive的方法能做到O(n)，但是需要O(n)的空间。巧妙的解法就是，注意到所有数的和是固定的，考虑切断后的非循环数组，找一段从首开始+一段从尾开始的两个子数组with最大和，等价于找一段子数组with min sum.

   总结，要擅长利用等价性转换问题，从而将新的问题转变为一个已知有好solution的旧问题。利用已知的经典问题来解决新问题，可以说是面试题目中相当重要的一个技巧

   2. largest rectangular problem：问题是这样的，一个N×M的棋盘，上面的数字要么是1，要么是0，那么要：a）最大的一个正方形全是1填充，b）最大的全是1的矩形。

   a）是用动态规划做，虽然方法也很好，但是这里就不提了。b）问题感觉上要比a难很多，为什么呢，因为rectangular比square有更大的自由度。不好用DP来做，分治也不合适。

   这题的奥妙就在于，利用经典问题。什么经典问题呢？其实是另外一道面试题，其本身也是有一定难度的题，题目是：给你一个统计直方图，假设每根柱子都是单位宽度，从图的最左边一个紧挨一个排到图的最右边，求在这个图里找到一个最大矩形，它不跟任何直方柱相交(边缘接触是允许的)。为什么提起这个题呢，故事是这样的，我之前没有做出O(N*M)解法的largest rect题，后来有一天遇到了这个直方图的题目，找到了很漂亮的O(N)解法，猛然回顾起那道largest rect的题，这次就很轻松的搞定了。

   3（鸣谢mittbbs jobhunting版上的一位面试官贡献自己出的题）有n个房间，小偷每天偷一间，偷的规律简单说就是随机行走，如果今天偷了第i间屋子，明天有一半的几率偷i-1，一半的几率偷i+1，注意如果刚好偷到了边界上，那么第二天只有唯一的选择。如果你是警察，你只能每天选择一个房间蹲守，并且贼的手段相当高明，偷了一个房间后，没有任何人能发觉该房间是否曾经被偷过。

   提示：奇偶性。总结：注意观察题目中隐含的性质。

   4. wild card匹配+搜索：假设你有一个dictionary(原题中是URL集合)，你要搜到到所有与 *a*bc*d 这样的输入所匹配的words。这里，*是通配符，可以当成是任意个任意字符(包括空)，怎么 预处理+搜索？如果输入是 ???a???b??cde 这类呢？ ‘?’代表单个任意字符。如果输入是? *的混合呢？

有道题目最近频繁出现，特地总结一下常规解法以及它的变体。有个经典变体我还没看到一个很经典的答案，希望有人补上，呵呵。

1. The largest rectangle under a histogram

http://www.informatik.uni-ulm.de/acm/Locals/2003/html/judge.html

Given: An integer array represents a histogram

Goal: Find the largest rectangle under the histogram.

Key observation: 输入为一个整数数组。如果某元素比它两边的邻居都小（比如Ai)，那么高度大于Ai的矩阵要么在该元素左边，要么在该元素右边，不可能穿过Ai。利用这个性质，想办法遍历所有的矩形。

Complexity O(N) where N is the size of the given array.

2. Maximum subarray with all 1’s. (Generalization of problem 1)

http://www.drdobbs.com/184410529

Given A two-dimensional array b (M rows, N columns) of Boolean values ("0" and "1")

Goal: Find the largest (most elements) rectangular subarray containing all ones.

Key observation: 从一边（假设右边）开始逐列扫描，构造直方图。每次构造出直方图来，用上面的解法求最大矩阵。每次构造直方图只需要O（M），解需要O（M），做N次。

Complexity O(MN).

3. Given a binary matrix, find out the maximum size square sub-matrix with all 1s. (Specialization of problem 2)

http://geeksforgeeks.org/?p=6257

Key observation:

假设输入二位矩阵为M,构造辅助矩阵S

```
If M[i][j] is 1 then

    S[i][j] = min(S[i][j-1], S[i-1][j], S[i-1][j-1]) + 1

Else /*If M[i][j] is 0*/

    S[i][j] = 0
```

Complexity: O(MN)

4. Imagine you have a square matrix, where each cell is filled with either black or white. Design an algorithm to find the maximum subsquare such that all four borders are filled with black pixels. (variation of problem 3)

1. write a iterative function to calculate fibonacci sequence
2. what is the difference of Delete table and truncat table.(about sql & database)
3. what is ACID.(about sql or database)
4. write a find() for BST, How to decide the element could not be found.((less than && No left child) || (large than && no right child))
5. how to reverse a link list using one point
(should I use recursive calling)

6. find the sizeof(int) without using sizeof(). （大家可以讨论一下，我用(~0x0)右移位做的，但是不对方希望的答案）

7. main()

```
{
   folk();
   folk();
   folk();
   printf("abc\n");
}
```

How many abc will be printed?

8. what is the difference between.
kill proces_id
kill -9 process_id

另外一家类似职位的电面，全部是stl的问题，很简单。

1. new a vector, what is the size(),
2. how to prevent relocation when push elements into a vector. (reserve())
3. why vector use relocation instead of chunks of memory.（keep compatible with C array）
4. how is deque implemented?(uses chunks of memory)
5. the different of set & map? what is the return value of find()? (an iterator of element)
6. what is the member function needed to use set.find() or map.find().(less( ))
7. what if the element class doesn't have less() operator? (pass a functor to constructor as comparing function)
8. for loop to find a element in a vector, what is the method the class need to support? (equal operator)
9. For smart pointer, How it implement the reference counter(a pointer to a int or unsigned int)

1. count the number of 1s in binary representation 我用的4位4位hash的方法 但是没有考虑负数情况，在面试官提示下，意识到对负数向右shift,发生的是sign extension, 所以这题要先对输入进行检查。
   2. remove duplicates from linked list, suppose you can use stl list. 我用的hash_set, 这题很顺利。
   3. 很长很长的string/file, 要我找出中间重复次数最多的character, character set 可能是任何char set, unicode... 我说mapreduce, 后来面试官说 如果是一台机器，8 个core, 1个process,怎么办。我说 似乎mapreduce需要在distributed system环境下(但其实我不肯定，如果一台机器多进程/多线程)是不是就也可以mapreduce）.面试官又给了提示，我才说multi-process,后改成multi-threading, 后来他问我那几个thread 怎么写协调，我说可以公用data segment in the process space, 但是要注意锁的问题。最后他问我说mapreduce和multi-thread的tradeoff， 我说communication overhead, 似乎他对我的答案满意。

   我和面试官说honestly第一题 我见过了 （我不知道这是不是画蛇添足了！！！！！）哎 总之 虽然觉得自己答得还可以 但是很担心！！

   第二个人先问了一些暑期实习的项目问题
   1. 设计一个counte class, 写一个函数算某机器最后一分钟处理任务的个数。我说用circular array, 他说right way to go, 但是在细节上澄清了一会。不知道是不是我的表达不够好。
   2. vector of vector of ints, 请implement bool hasnext()和int next() 我说要用

一个变量track在nested vector中的位置，他提示用2个，然后我开始code, hasnext code完了 然后也解释给他，似乎没有问题。next() 由于时间不够，但我把逻辑解释完了，把关键的那4，5行code出来。

因为赶着年底毕业，九月底才开始投简历。这个offer来的太快，小startup就是动作快，从十月初联系我，到 offer, 就两周。那几个大公司的 on-campus interview还都没开始。也算是 hot startup,但这里肯定没人知道的，移民版知道这个公司的更多些。就不透露公司的名字和考题了，见谅。
HR 联系之后，先是组里的头直接电面，问了一个他们实践中的问题，我没想出答案，但还是扯了扯。后来就在谈公司做什么，我有准备，问了很多问题。刚放下电话,HR问我什么时候作 coding test, 可以马上把题发给我, 就是fill Java class, 实现某些功能，一般给 2-3个小时。我想这么大工作量，不能拖，否则牵扯时间和精力，就说马上作，决定不准备了，冒一定风险。结果一个小时做完发给他们，小impress了一下。然后另一个 team lead 马上打电话二面，顺便考察一下是不是真是我自己写的程序。这就是一天三面，两电一编程，然后就给 onsite了。面了组里的4个lead,HR 和 Founder。前两个基本都在问我的 research和 big picture, 没有考我，第三个问了两个经典算法，属于偏简单的，版上都有，出现过很多次的。有一个我当时不知道，提示着才做出来。感觉面试的时候脑子不是很转的开，没有准备的话，很多题都不可能马上想出来。反过来，有准备的，就当不知到结果，从头把思路说一遍。吃午饭的时候，拍了拍马屁，开了开玩笑，到下午果然放水。第一个人就问我 PhD 为什么不考虑 academic job,这个之前 HR也问过，基本是 PhD必问的 behavioral question, 一定要准备。我就说整天synthetic data, abstract problem 不如实际的问题 motivating.
然后就是我在问他做什么东西，追的很 detail, 就没他问我的机会了。第五个还是那个联系我的 HR, 人很精干，办事效率高，问了很多behavior的。 你最想作又没机会做的，我就说 leadership。 还有什么你总做不好，例子是 time management, work life balance, 我说了一些 research的艰辛。还有你跟人关系处的不好的例子，怎么解决的等等。最后是大老板，Founder, 向我夸赞他们的产品，策略和公司影响，还有公司的光明前景，我们组的重要性，还有股票来者有分，听得我直流口水。Onsite 后第二天晚上就给 口头 offer了。要了 reference，background check, 第四天正式 offer,但只给我一周答复实践。觉得工资低了点，就大胆照着Google的 negotiate了一下，涨了一些，达到了 Google PhD的下限，觉得知足了。我这样的，去面 Google, 估计不到一成的胜率。
再说其他面试经历，年初有一个猎头从 linkedIn上找到我要给我介绍什么 DE Shaw, Two Sigma. 我说年底要毕业，thesis 还八字没一撇，到时候再联系吧。到了九月底，联系那猎头，很热情的要了我的简历，还帮我改了改，说很好，我马上给你投 DE Shaw , Two Sigma. 没两天就受到裸拒，深受打击。后来一看，也不冤。这是两个名hedge fund 要不牛校(MIT, Stanford, Berkley, Ivy 级别的)，要不拿过竞赛奖的，否则基本不会考虑你。恍然大悟，原来这个该死的猎头本着中六合彩的心态拿我撞运气。
从此发现了可以走quant dev/strategist这条路。从九月底，先投了两个 wall street 的。 第一个是作 CDO相关的 quant strategist, 过了coding test, 和两轮 phone interview,然后问我什么时候来纽约，好安排 onsite. 我说你得出钱呀，被告知他们一般只招local,早说呀。费这么半天劲，就当练手了。另一个是 hedge fund 的 data specialist, 通过另一个猎头找的。面试我的是中国人 team lead。由于被 DE Shaw, Two Sigma拒的神经错乱，整天想着牛校，在结束之前竟然问人家是什么学校出身，被不愉快的谢绝回答。以为就完蛋了，猎头打来电话说不错，继续。 二面又是一个中国同胞，感觉比一面还好些，最后又问人家是什么学校出身，不长记性，又被不愉快的谢绝回答。至此就没消息了。肯定是觉得我人品有大问题。如果二位在版上混，请不要见怪。
这两个公司，除了经典算法题如longest increasing subsequence, linear time selection, verify if S1 is substring of S2, 还都问了很多 C++， Java 概念题，

如虚函数，如何用 destructor 实现 exception safe code,顺便提及为什么Java 不需要 destructor？因为 Java 有 garbage collection, 但是 Java如何 safely release external resource in case of exception? 答案是用 try/finally.用finalize 也可以，但不是同步的（asynchronous）。 还问了 smart pointer, reference counting, Java garbage collection, 这些都是相关的。也投了一些其他的金融类，都默拒了。看来还是找猎头好，至少不会默拒。

上周和这周on campus 面了 Microsoft, Epic, Amazon, Facebook

Microsoft: 从 numeric array 里找两个数之和最大。刚开始还以为是经典的找和为固定值的两个数，其实这个更简单。找最大的两个数就行。Simple O(n) algorithm. 都没有让优化到 N+lgN-2 个 comparison. Instead, 让考虑意外情况，比如数字太大，加起来overflow怎么办.

EPIC: 竟然没有technical question ！（干什么来的？）, 跟我套了套近乎，说他N年以前选过一门database的课，跟我写在简历上TA的是同一门课，同一个professor.我说现在教的很不一样了，顺便吹了吹。继续说到他们在用一个很老很奇怪的database叫什么 CHRONICLE 之类的，70年代他们 CEO自己写的，现在还在用。还属于post-relational database,我后来在Wiki上也没找到这个什么 Chronicle.又说他们还在用VB,小惊讶了一下。后来在Glassdoor上看了一下EPIC的员工review，很多人在抱怨working on legacy code, old technology … 尽管EPIC在 healthcare 里占有很大份额，但得不到什么技术上的提高，作码工还是那种最累的，从 H1B visa 纪录看，pay的也不是很好的样子。劝大家能不去就不去。一家之言。

Amazon: 这回是经典的从 array 里找两个数，和为一个特定输入的值。一种是用hash, expected O(N) time, 一种是sort之后从两头往中间scan, O(NlgN)。第二题是verify that a binary tree is a binary search tree. 就是 recursive in-order traversal, 然后看是不是都是单调递增。如果不用多余空间，只要再用到一个全局变量记录上一个访问节点的数值就行了。当时脑子短路，提示了一会儿才想出来。 这两个都要再纸上写code,他们存档讨论

Facebook: 把一个binary tree变成double linked list。 也是写一个recursive in-order traversal, update pointer 的时候稍微有点tricky. 第二题是经典的计算 x/y 不用 division operator. 我说用 log 和 exp, 他说可以。又问这些也不能用怎么办。 提示了一下 x, y 都是正整数的话，从 x减去y 一直减到0,看要减多少次，很obvious 是 bisection 了。面完之后说我肯定第二轮 on campus了,如果过了就是onsite。后来recruiter打来电话说根据我问的问题，他们决定让某个比较match的组约电话 interview时间,马上会联系我， 而不是继续下一轮on campus。等了一天还没消息，不知到是不是传说中婉拒的最高境界。

这些大公司都只过了一轮，题也比较简单，见笑了。手头 offer催的紧，觉得不算太差，base pay 的和google也差不了太多，potential bonus 比例还高些，但要看公司业绩，所以很可能是望梅止渴。Stock给的也没google值钱, 不过YY起来还够用。还要赶着写论文，就决定从了。

另外说一下Facebook现在给的stock确实很诱人，我委婉问interviewer他有多少股 (2年员工)，是不是很有钱了。 感觉他有上10万股，但很frustrated的样子说，不IPO就跟废纸一样。也许是怕我嫉妒他。听说二手市场买卖facebook的share有一定的成交量,但不知到是不是有限制，所以伦不上小喽啰门。

最遗憾的是没有试试Google。找好了递简历的同学，但是一直觉得没准备好。想放到最后。估计现在去了也白给。手头只有一个 offer 就从了，但从各方面分析，也只能这样了。现在只剩一个月写thesis了，八字才刚有一撇:(

Yahoo 电面 印度人

1.电话键盘上1-》abc 2->cde... 现在来一堆数，未知长度，比如123456..... 请输出序列可能对应的所有字符串

比如 123 输出acf, acg, ach, bcf...

2.检测链表是否有环
3.sql语句
employee(id(primary key),name)
employee_bonus(id(primary key), bonus) （现在觉得这题 似乎有点问题，因为他和我说 id可以对应多个bonus, 那这还算是primary key吗。。。）
请写sql 输出name和这个人bonus总和。

MS on campus interview - first round
1. 简历问题
2. 给一个字符串检测是否是valid ip address, 这题他似乎是想看看我的思路，我说regular express, 他说要code, 我就写了一些，解释了一下。总之，这题真要追究起来，细节颇多，但因为我们每个人都只有30分钟，所以他没有要求完美的答案。

我知道的其他ms校园面试题
1. bst删除一个节点
2. 矩阵，每行从左到右递增，每列从上到下递增，找一个数。

Amazon所有轮的题目
1. 数一个字符串中的单词数。
2. 经典的 一堆数只有一个出现2次 其他数出现一次 怎么找出那个重复的数。请给出4种以上解法。。。包括brute force...
3. longest palindrome.
4. design restaurant reservation system.
5. atoi, 只能用一个loop. 要考虑-213， 2-13， a2-13....各种情况。。 这题看似简单，但是要小心，容易出错。
6. design a phonebook, 要求输入人名的时候电话随之更新。比如a给出一个电话，abc给一个，abcd再更新给出一个电话。trie的implementation, 假设每个节点存一个数据和26个指针。
7. c++中static和virtual keyword, 请解释用法。
amazon给我的感觉就是算法不一定很难，但喜欢考oo design.

1. 一个rotated的排序整数数组，比如A=[6，8，1，2，4，5]，写code找一个给定元素，并分析复杂度。其实就是binary search的变体，但是需要考虑两种A[m]中值的情况加以判断。
   2. 他谈到facebook的log，如果每个log文件有10 billion行，每行包括timestamp, user_id, visited page三个field。如果高效的统计一个月内用户访问量最多的十个网页。假设文件已经按照timestamp排好序了。不用写code，谈想法就成。我的解法是，如果内存足够，建立一个<网页,访问数>的hash表，每读入一条记录，该网页key对应的项加一。但是加入内存不够大，可能需要map reduce方法。面试官马上说，内存根本不是问题，用个4G或者8G内存来算这个月访问量很值得，不要省内存。然后我说，那hash表就好了。面试官表示赞许，接着提出引申问题：
   3.假设我们已经算了一个月的，如果这个滑动窗口要往前移动一个delta量，就是说，已经算好timestamp=[t_1..t_x]区间的访问量，要计算timestamp=[t_2..t_x+1]的访问量，提出高效算法。当然，最brute force的方法就是针对[t_2..t_x+1]再建一个hash表，显然这不是面试官想要的。我提出来，可以建个multi-level hash表，外层为网页，内层为timestamp，但是仔细想想也不算好。面试官也没说他们的解法是啥。

1 external sort
2 一道正态随机的题目。我到现在还不太明白。。。。
3 print BST in level order
4 实现linkend里查找两个人之间connection的功能。（如果每人有100个熟人，假设任

何两个人之间只隔6个人，需要space 100^6，内存放不下。所以改用同时从两边bfs, 需要space 2*100^3）
5 合并两个直方图，新图是原来两图的和。（直方图用点的array储存，比如一个直方图有两个矩形:x 2-3, y 4; x 3-5, y 3.表示成{(2,4), (3,3), (5,0)}.计算的新图点对就类似成combine two sorted array了）

apple网投，一个月后phone。
1 c++的多态有哪些实现方法。（继承，template specification，好像还有一个，忘了。。。。）
2 为啥用mutex(to avoid unstable states)

ebay网上投简历，1轮phone interview+onsite. onsite从9点面到6点，一个小时一个人，累死了。。。。签了保密协议，具体题目就不说了（能记住的只有下面这些了。。。）。
1 design patten. factory method, factory, visitor, bridge, class design.
2 反转link list
3 external sort
4 实现查找两点间最短路径
5 1堆数，找出所有和为定值的三个数

```
#include <stdio.h>

void permutation(char * p_str, char * p_begin)
{
   if(!p_str || !p_begin)
   {
      return;
   }

   /*
    * If p_begin points to the end of string,
    * this round of permutation is finished,
    * print the permuted string.
    */
   if('\0' == *p_begin)
   {
      printf("%s\n", p_str);
   }
   /* Otherwise, permute string. */
   else
   {
      char * p_ch;

      for(p_ch = p_begin; *p_ch != '\0'; ++p_ch)
      {
         char temp;

         /* Swap p_ch and p_begin. */
         temp = *p_ch;
         *p_ch = *p_begin;
         *p_begin = temp;

         permutation(p_str, p_begin + 1);

         /* Restore p_ch and p_begin. */
         temp = *p_ch;
```

```
            *p_ch = *p_begin;
            *p_begin = temp;
        }
    }
}
int main(int argc, char * argv[])
{


    char strr[4]="123";
    char strd[4];
permutation(strr, strr);
    return 0;
}
```

UTF8解码器

线段overlap

死锁机制

fair streaming sample

最不好的是,忘记可以用bitmap记录sparse data,选择了hashtable.

最近开始准备OO design questions, 这两天学习了一下design pattern, 用Amazon了

经典的Hotel Reservation OO

Design的问题练习了一下。希望能和版上各位高手交流一下。

(1) 问题1: 这个system应该就那些类

在version 1设计，我定义了如下类Room, SingleRoom, DoubleRoom, SuitRoom,

Hotel, Customer, Reservation, Date。 因为感觉这个design,主要就是如何设计check different types

of rooms' availability和make reservation。

(2) 问题2: 这些类的关系应该如何

其实就是"has-a" and "is-a" 的设计，也就是用inheritance或是composition

。根据直觉定了了如下的UML图

以上两步估计大部分人都能得道，不过如何才能打动interviewer呢？怎么才能提高这

个设计呢？

我想可以从两各方面入手：（1）program with interface (不过这个好像是java

programmer的rule) （2）多用design pattern, 因为design pattern能decouple code,这样就能提高code的

resuability. 在什么情况下，如何使用design pattern就是门学问了，需要经验阿！

不过对于这个问题，我想了些改进方法。

(3) 问题3：如何create object

用factory method pattern来建造不同的Room object, 虽然目前的design只有三个 Room-based classes,不过随着以后系统复杂度的增加，可能回有n个Room-based classes。我猜想design的时候考 虑日后扩展的flexibility，应该也是interviewer喜欢看到的把。

(4) 问题4： 如何hold多个Room-based objects

是用一个list来hold所有的SingleRoom, DoubleRoom, SuitRoom的object,还是用三个 list来hold各自的object? 这个没有想好，目前就用了一个list, 因为估计总room的数目不会太大，对单个list进行 iterate的效率不会太差。

(5) 问题5: 如何有效的check availability.

假设有一个customer要查询每种room从某天到某天的availability, 那么需要对所有的 Room subclasses都定义一个查询函数checkAvailabity()。这样的话不是很好的design。因为日后要加其他的查询： 如Room的朝向（如，有没有view之类的:-)），那又再改interface, 然后对所有的classes添加函数。麻烦！

所以， 我考虑用strategy pattern可以日后extend各种查询。

(6) 问题6： 找到availability, 如何设计reservation

回想一下，make a reservation其实需要多个步骤的，需要填用户信息，需要填信用卡 （如果是pre-pay的话）， 需要计算tax, 还可以用discount code的，最后得到total amount。这些步骤都是 sequential的，有一定的次序。所以我们可以用builder pattern来generate所有这些步骤。

写个code（看下面link），欢迎各位排砖。

http://www.ideone.com/KsDCV

--

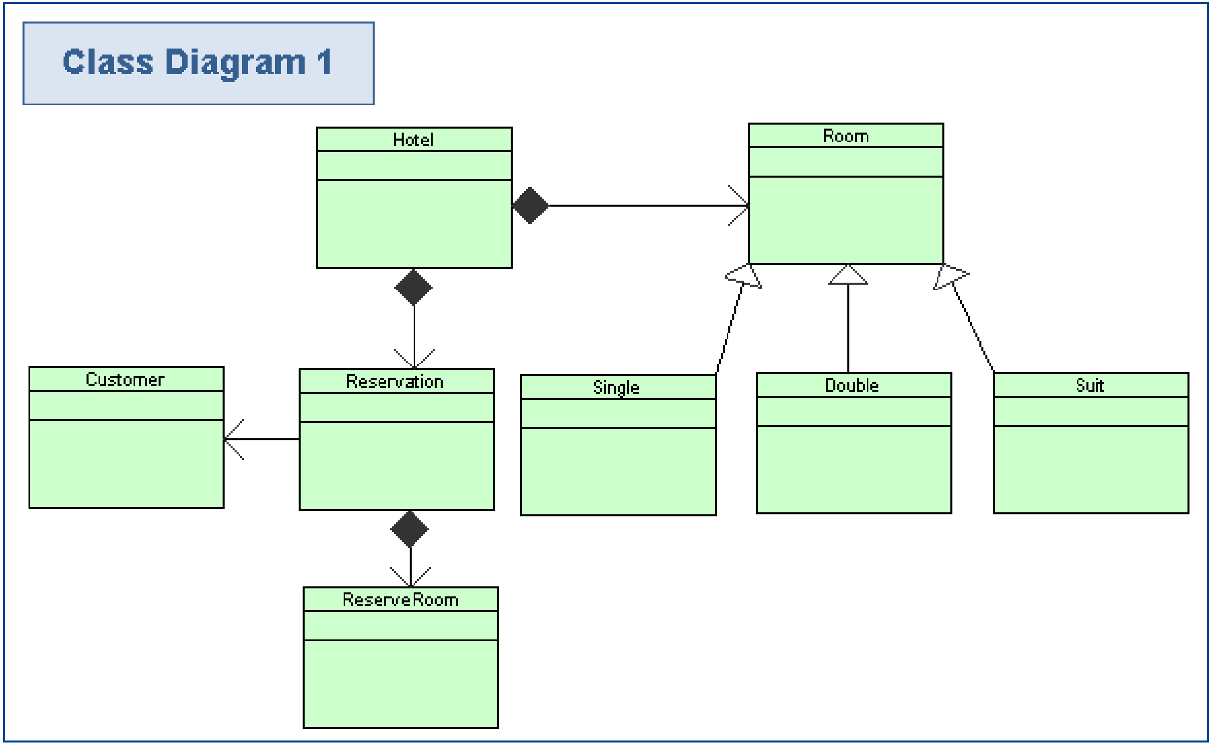
※ 修改:·langqinren 於 Nov 13 10:29:29 2010 修改本文·[FROM: 71.62.]

※ 来源:·WWW 未名空间站 海外: mitbbs.com 中国: mitbbs.cn·[FROM: 198.82.]

此主题相关图片如下：

HotelReservation.gif

(5967 字节) [删除]



Phone1：
Behavioral: Your biggest challenge, do you know our product?
Tech: 经典的html里找email的题 (using regex)

- 找anagram
- Deck shuffle algorithm
- Two stacks for a queue

Phone2：

- N-way merge和时间复杂度 (n-way 和 2-way的比较)
- 手机输入提示功能 (trie)

两个phone都不难 很快拿到onsite 同时面两个组 onsite发现 A组全是白人 B组全是阿三 结果被阿三给放倒了...
Onsite1 - 老美 A组manager： 问了问profile, 给了一个oop design的问题 不是常见的电梯或家具题, 完全是他们所做项目的设计. 这题回答的一般, 最初给的答案不是他想要的。。。在提示后 完成设计

Onsite2 - 阿三 可能是Bar raiser, 因为不是A组或B组的人

1. 超大文件 查找unique数字 (hash, bitmap, 外排)
2. Load balancer问题: 一个balancer, 两个servers, 如果其中一个server down了, 问如何设计balancing algorithm使得用户感觉不到延迟(e.g. 如果使用round robin的算法 需要从server A返回不available的信号, 再转向server B, 就会有延迟）要求只有两个servers available. (这题给了两个答案：read/write through cache, 和recent response time 都被否决 不知道想要什么)
3. 一个balancer， 多个servers 问如何设计 回答说master-slaves + hashing (没有说consistent hashing 因为想弄明白他的问题). 结果果然没有问consistent hashing, 问如何优化slaves的storage，回答sharding。 这题不置可否，讨论了一会时间到

Onsite3 - 老美 水过

1. linked list 找random k元素 (reservoir sampling)
2. 又是anagram

这个老美感觉是来应付交差的。。。 后面问了一堆behavioral question...

Onsite4 - 阿三panel B组manager和B组成员
B组全部都是阿三。。。 问machine learning的问题 都是和Amazon实际问题相关 没有准备。。。回答的差

Onsite5 - A组 东欧人

1. C++多继承转Java code (using combination)
2. user activities log, 找出最长的session (一小时之内的activities被认为在一个session里). 分别问了单机 多机的情况, 然后coding

deep copy linked list: 俩pointer, next, random
Linked list merge sort
write iterator for a list of arraylist
最自豪的事儿
找到一张图片里面所有的word 的bounding box
比较两个directory是不是完全一样
OO design for chess game,design render loop
number of 1 in binary representation for integer

find kth to last item in linked list
好多behavior 问题
怎样实现traceroute
tcp vs udp
caching, indexing in db, load balancing, sticky session etc.
怎样优化web service,找bottle neck

Round 1:

1. C++ Copy Constructor, 包括接口和deep vs. shallow的区别，这题答得一般，有些细节没搞得特别透彻

2. Hash Table的实现和技术要点 (仔细看过一遍wiki就没问题了）

3. Large file, multiple lines, how to get any line in equal probablity. 这题可以问得很深入，比如文件太大内存无法装入如何办。

我回答的思路：

内存够文件内容就都装入内存，然后randomGenerator选一行,one pass

内存不够可以记录每行的偏移值在内存，这样之后可以fseek到那一行去读取

如果偏移值都放不下，可以divide into ranges, 当然这个range的stepsize不好选，可以预估，也可以动态改变（到这一层其实大致给些思路就ok了）

Round 2:

1. Research problem

2. How to build a service to generate an unique number for each client request, 这个问题也可以问得很深入。

我回答的思路：

systemTime + IP --> IP不唯一 --> systemTime + ConnectionID

multiple requests --> build a distributed service so we need add hardware signiture to make sure the unqiueness

Another solutions is building a hashTable to store all generated numbers for a check before sending number to client

拿到Onsite ...

Microsoft:

Some behavior questions

Singleton pattern concept, write a Singleton class (in multithreading environment)

拿到Onsite ... (特别感谢一位谈不上很熟的朋友给了很多帮助，所以这个onsite拿得最顺利)

a startup:

String2int, 这题不难，但要考虑很多细节，比如+/-号，溢出等

Hash Table 实现与技术要点，与stl:map的区别和优劣

finding missing integer (enough memory solution, not enough memory solution)
, 基本上就是用bit-vector的思路，如果内存不够就1bit to a range

拿到Onsite ...

Google

Round 1:

1. 矩阵乘法 （紧张导致脑袋不清晰，虽然最后代码正确但中间出了挺多纰漏）

2. 一些基础的数据结构，arrary, list, hash, tree等等

虽然自我感觉糟糕还是被容许第二次电话面试

Round 2 （一个语气挺tough的同胞）：

1. 股票的买入卖出，max the profit, 如果有多个pair产生max profit如何处理（这题因为自己编程做过所以答得应该挺好）

[回复] [回信给作者] [本篇全文] [本讨论区] [修改] [删除] [转寄] [转贴] [收藏] [举报] 0 0 [1]

发信人: linkaixin123 (林开心), 信区: JobHunting

标 题: Amazon电面 (面经)

发信站: BBS 未名空间站 (Wed Nov 24 16:42:46 2010, 美东)

报个面经，攒一下RP。

几个问题，都是经典题，另外考了不少Java的基础知识，还有一道设计题。

== Java ==

1. interface vs abstract class

2. the usages of 'final' keyword

3. overload vs override

4. concept of 'equals' and how to override (need to override hashCode() also)

5. checked vs unchecked exception

== Misc ==

1. 50,000 HTML pages, extract the ones with phone numbers (regex)

2. Find the odd-occurence number in an array (I told him that I have seen the problem before and he asked the next one)

3. Find the duplicate number in an array of N ints (1<=x<=N-1)

4. Design a transportation system (warehouses, routes, trucks, fuel, MPG, etc)

== Homework ==

Write a Card Game program and send it to him before evening.

1. the difference of new/delete malloc/free. what if new/free, and malloc/delete?
2. sizeof an empty class
3. sizeof an empty class with virtual function.
4. can we call virtual function in constructor/destructor?
5. what is pure virtual function?
6. How to disable class initialized on heap,
7. How disable class initialized on stack,
8. where static varible saved, heap or stack?(bss, data)
9. what does static mean in C++
10. what is 4 default functions for empty class
11. when throw exceptions, why use reference to catch exception(这题
12. what is the syntax to catch all exception?
13. can constructor/destructor throw exception.
14. what will happen if 2 or more exceptions happened?
15. why we need exception in C++?(what is the benefit we can have by using exception)
16. What the design pattern do you use?
17. Tell me all STL containers you know.
18. Can we use auto_ptr as element in set or map?

== Data Structures & Complexity ==
2. Lookup in linked list and array (sorted, unsorted)

3. Sorting strategies (comparison-based & non-comparison-based)
4. Lookup, insert, delete in hash table.
5. How to resolve collision (chaining, open addressing)
6. How to support delete with using open addressing
7. What affects collision (hash table size & hash function)
8. What the complexity when using dynamic array for hash table?
(insert 1M elements takes how many steps when we always copy over => 2^n (geometric serie))
9. What is BST, balanced BST? How to maintain balance with inputs like 1, 2, 3, 4, 5 ... (red-black tree)

== OS concepts ==
1. What is process & thread? Difference between kernel thread & application thread.
2. What is virtual memory? What is it good for?
3. How to implement thread? (save/restore registers during context switch)
4. Write a program to check if stack grows downward or upward. (A calls B and prints the addresses of two local variables)

== Design ==
1. Design an web-site for playing card games. (what classes, what are the responsibilities, users vs players)
== OO ==
1. What is inheritence and polymorphism?
2. How to overload a function? Why return type is not enough?
3. Is-a vs Has-a
4. aggregation vs composition
5. Strategy design pattern
6. Issues with multiple inheritance (diamond problem)
7. Class member vs instance member
8. Implementation inheritance vs interface inheritance
9. How to reuse code? (inheritence, template, library, external program, web-service, FPGA, etc)
1st:
1. 讨论我的博士研究项目
2. 如果SNMP agent不能获取数据，或者获取的数据不符合预期，如何诊断该问题？
3。我做过的最有挑战的项目是什么？
4。用邮件写代码,然后讨论我写的代码：

```
unsigned char * get(int sizeOfArray, int sizeOfRecord);
void release(unsigned char* ptr);
```

该函数可以实现：

```
unsigned char ** array = get(5, 10);
snprintf( array[0], 10, “hello world\n”);
snprintf( array[1], 10, “hello again\n”);
```

5。Java的基本概念

2nd
1。Apache的log file如何找访问量最大的网页 （用linux shell写个小script）
2。如果某网站访问量突然增加，可能是什么情况发生，如何确定各种情况（1。暂时的Popularity激增 2. DDOS Attack 3. 网站添加新的内容）
3。Java基本概念+设计扑克牌的类
4。读reverse string的代码（基于stack和对换位置）

Onsite Interview

1. 很高级别的一个manager，介绍group, 各种behavior questions, 无任何技术问题。我早上8:00开始interview的，估计manager还没想好题，或者不像一大早就为难我把：D

2. Bar raiser; 如何实现phone Book（我的答案是trie,）, 并要求写出insert函数；外加一推java的基本概念

3。给一个maze：for example:4 0 5 6 1 0， 每个number代表最多可以走的步数，问是否能从a[0]走到a[n-1]; 问如何判断Web Service 运行正常，怎样解释response time的variance, 谈谈botnet

4. 网络题：MTU Discovery, Switch&Router, IP header, VLAN怎样实现的，路由表怎样实现的，bitmap，hashtable. 还写了一个很简单的程序

5。HashMap如何实现的；

Userid	PageID
A	1
A	2
A	3
B	2
B	3
C	1
B	4
A	4

找出最常用的3个访问序列：

对于用户A：1-2-3， 2-3-4

用户B：2-3-4

2-3-4 是最常见的

写atoi的程序

设计rent movie的类

6。lunch with hiring manager. 我对该职位的理解，为什么感兴趣，如果加入team会如何做啊，还有QA部分； 饭后问了到网络架构题

第一轮是一个印度大哥，PhD的研究方向（machine learning)跟我很类似。问了：

1. 如何向不懂machine learning的人解释LASSO和l_1 regularization。

2. 解释什么是PCA；为什么二维数据中PCA找的主方向和做线性回归的fitting不一样。

3. distributed median，（unsorted）数字分布在几台机器上，设计分布式算法找到它们的median，要考虑网络通讯的overhead。（先local sorting，然后一个数一个数去看是不是median就行，结果我设计了一个巨复杂的recursive algorithm...还好那个印度大哥比较nice）

4. BST中任意节点的下一个数字。这个大家都会吧，呵呵

第二轮是中国人，后来查了linkedin发现是当年本科学校的传奇人物。

1. 在整数中找3个数，使它们和为0

2. 怎么设计一个并行的linear regression solver

3. 什么是decision tree, 怎么去建一个DT (用Gini墒判断用哪个attribute），什么是Gini墒。

4. maximum area under histogram （忘记了O(n)算法，简单说了思路）

设计一个web cache server，假设存储网页数量是10个billion，打算怎么设计

1. Is everything in-memory possible?

2. If one machine is not enough, how to partition? What's the partition key?

3. What's the index strategy? Hash based or tree based? If use hash based index, how to handle the hash-collision? (I remember Database textbook has

discussion on this)
4. If memory is not enough, how to handle?
5. What's the complexity to check whether a URL is in cache or not? (is bloom filter useful?)
6. Do we need a meta table on which table/machine is where for what?
7. May also need to consider Replica for reliability.

Big table (in memory version) may be a good reference here.

1、1~1000的数中有一个重复的数，把这个数找出来。
2、找出Binary Search Tree树的Median number。
这两道题的要求都是提出尽量多的算法，并给出复杂度。
3、第2题我说了一个算法：
如果事先知道每个节点的子节点个数，假设节点总数n是奇数，m=(n+1)/2，根的左子树的节点数是k，右子树(n-k-1)。这样的话，就把找中数转换成找第m小的数：

```
if (m == k+1) return root;
else if (m <= k) find_m(root->lchild, m);
else find_m(root->rchild, m-k-1);
```

我说复杂度可以达到O(log
n)（不包括预处理的时间。现在想一想不准确，请各位帮忙看看这种算法对不对，如果对的话复杂度是多少？

然后他就让我比较几个时间复杂度：O(n)、O(n^2)、O(log n)、O(n log
n)，问哪个是最快的。是不是任何时候O(log n)都比O(n)的算法好？他想要的答案大概是："Big O notation discards multiplicative constants on the running time, and ignores efficiency for low input sizes, it does not always reveal the fastest algorithm in practice or for practically-sized data sets."
http://en.wikipedia.org/wiki/Big_O_notation
4、写出创建我说的bst的过程的代码，即二叉树中的每个节点都包含其子节点的个数。
我当时我一个地方写错了，一个参数没有加引用，但是他当时也没看出来，对我的答案还比较满意。

第一个，聊了很久ERP方面的东西，最后只问了一道技术题。

An array of continuous integers, only one number is duplicate, find out this number. What about there is a number is missing, how to find out both the duplicate and missing numbers.

第二个，问了heap跟stack的区别，以及什么情况下会OutOfMenoryException。然后就是技术
题了。

Reverse a string. e.g. he is a man => man a si eh
Reverse a string without reversing the words. e.g. he is a man => man a is he
Find the square root of a float number.

午饭后第三个，同样聊了很久我现在做的ERP方面的东西，问了下5年的career path和遇到的
most challenging technical problem in previous projects, 然后就是技术题了：
1) Give three database tables, Invoice, InvoiceLineItem, and PartMaster.
Find the design problems. 主要就是要变成一个normalized design. 要问清楚具体的business requirements，然后针对这个requirement进行修改.
2）Write sql to find the most recent invoice date for all customers. 我记得是这个

```
SELECT customer_name, max(invoice_date)
FROM Customer, Invoice
WHERE Customer.customer_id=Invoice.customer_id
GROUP BY customer_name.
```

对了，Customer table是normalized后得出的。

3) Given a list of methods as follows.

```
Station radio.getStation();
void setToNext();
void setToPrevious();

String station.getFrequency(); // Station的Unique id.
int station.getStrength();
```

Find the top 6 strongest signal stations： List findSixStrongestStations(Radio radio){...}

第四个，A list (LinkedList) of n numbers, p1->p2->...->pn. Find k random numbers.
1) n is known
2) n is unknown

Given a BST, find the successor of a given node. The node class is as follows.

```
class Node{
Node lc;
Node rc;
Node parent;
}
```

最后一个是director, 聊了一下他们的ERP的架构，使用的语言。说了下这两组的区别和主要进行
的项目。最后一个技术题。

Given a file's full-path, e.g. c:\foo\bar\dir12345
given a positive number n, find an abbreviated path from the full lpath whose length <= n with the following conditions:
1) Drive must be included, i.e. "c:" should be in the result.
2) The suffix part has higher priority to be included since it contains more important information.
3) Each part can not be partially included in the result.
4) The skipped part represented by "...".

给一个array of integers, 需要多少个comparison能找到min? how about min and max? what's the best worst-case? (3n/2)

怎么merge两个sorted的array，n 个呢？

给一个tree，把同层的nodes都连起来，assume每个node都有个指针，你要让它指向下

一个同层的node。

count 1's in an integer.

mergeSort, mergeSort without recursion, why nlogn?

Find whether one string is a subset of another string (not need to be contiguous, but order should match).

Print the nodes on the exterior of a binary tree in a anti-clockwise order,
i.e., nodes on left edge, then leaf
nodes, then nodes on right edge.

Find two integers in an array that sum to a target integer.

【谷歌（Google）】——谷歌笔试面试题集锦

发布者 ：admin 时间 ：[2010-04-27]

一、笔试题

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->uvSvu|w所识别的语言是：（）
a、uvw*vu b、（uvwvu）* 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<<end1;
```

当n=11时，输出：（）
a、12 b、13 c、14 d、15
5、下面哪项不是链表优于数组的特点？（）
a、方便删除 b、方便插入 c、长度可变 d、存储空间小
6、如何减少换页错误？（）
a、进程倾向于占用CPU　b、访问局部性（locality of reference）满足进程要求)
c、进程倾向于占用I/O　d、使用基于最短剩余时间（shortest remaining time）的调度机制
e、减少页大小
7、80x86中，十进制数-3用16位二进制数表示为？（）
8、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

9、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

10、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 ", foo(3, 5));
```

a、81 b、27 c、9 d、3 e、1

11、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

12、以下算法描述了一个在n国元素的双向链表中找到第k个元素的
方法（k >= 1且k <= n）：
如果k <= 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、θ(nlogn) b、θ(max{k, n - k}) c、θ(k + (n - k)) d、θ(max{k, k - n}) e、θ(min{k, n - k})

13、有一个由10个顶点组成的图，每个顶点有6个度，那么这个图有几条边？（）
a、60 b、30 c、20 d、80 e、90

14、正则表达式L = x*(x|yx+)。下列哪个字符串不符号L（）
a、x b、xyxyx c、xyx d、yxx e、yx8

15、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

16、Fibonacci,求f(4）使用递归调用f(1)的次数（）
f(n) = f(n-1)+f(n-2), f(0)=0, f(1)=1
a、5 b、4 c、3 d、4以上

17、有关哈希表正确的说法（不定项）（）
a、哈希表的效率和哈希函数。。。。相关
b、哈希表的解决冲突方法慢，回影响哈希表效率
c、使用链表哈希可使内存紧凑

18、下列排序方法最差情况时间复杂度为O(n^2)的是：（）
a、插入 b、归并 c、冒泡 d、快速

19、写一段程序判定一个有向图G中节点w是否从节点v可达。（假如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
}
```

20、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。

21、以下函数的结果？

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

22、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);
}
```

23、T(n) = 25T(n/5)+n^2的时间复杂度？

24、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？

25、正则表达式(01|10|1001|0110)*与下列哪个表达式一样？

26、实现两个N*N矩阵的乘法，矩阵由一维数组表示。

27、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。

28、打印出一个二叉树的内容。

29、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。

30、求一个二叉树的高度，如果只有root结点，高度为0。

31、将稀疏疏组中的非零元素提取出来，用链表表示。

32、两个n维数组，已排序，为升序。设计算法求2n的数中第n大的数。要求分析时间和空间复杂度。不用给出代码。

33、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。

答：最大子序列

问题：

给定一整数序列A1， A2，... An （可能有负数），求A1~An的一个子序列Ai~Aj，使得Ai到Aj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20。 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Sum(i, j)是A[i] ... A[j]的和，那么Sum(i, j+1) = 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子段进行分析，若有比当前max大的子段，继续更新max。这样一趟扫描结果也就出来了。

二、面试题

1、求直方图的最大内接矩形，假设每个细条的宽度为1，这个题很hot，两个人来问。我没想出什么好的算法。
2、NxN行列有序的矩阵查找一个数。以前有人遇到过。O(N)的时间复杂度。
3、给定一篇文章，求包含所有单词的最短摘要。O(N)的时间复杂度。
4、将MxN的矩阵转秩，要求O(1)的空间复杂度。参考群论中cyclic group,group generator
5、开放式问题，怎么避免重复抓取网页。
6、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。
7、写一个singleton pattern的例子。
8、vector vs. arraylist, growth strategy & complexity。
9、在C++文件中只declare class A, 但不以任何方式define class A, 是做什么用。
10、virtual function。
11、讨论html vs. xhtml vs. xml。
12、描述在浏览器中敲入一个网址后所发生的事情.dns,cache等。
13、给一个长度为n的整数数组，只允许用乘法不允许用除法，计算任意(n-1)个数的组合乘积中最大的一组。。。写出算法的时空复杂度。

1. Int to string (pay attention to the smallest negative number)

2. Given two sorted list, find the k smallest number (binary search)

3. You can win three kinds of basketball points, 1 point, 2 points, and 3 points. Given a total score X, print out all the combination to compose X. (recursion/ Dp)

4. Given a stream of integers, at a given time, there is a number appeared more than half time, how to find this number.   (classic streaming algorithm)

///面试书recruiter推荐
1) Mark Joshi: "Quant Job Interviews: Questions and Answers".   I have heard very good things about this book.
2) Xinfeng Zhou, "A Practical Guide to Quantitative Finance Interviews"
个人觉得非常有用， 大部分问题都在这两本上。
算法，C++, stochastic calculus 就看比较标准的几本。

- sqrt(i)=?

- You and me roll a dice,first one gets a six wins. You roll first. what is the probability of you winning?

- A stair of n steps. Each time you step up 1 or 2 steps. How many different ways are there to reach the top? what is the asymptotic limit?

- Moment generating function of standard model.

- Write a simple function to return Fibonacci number F(n). how to avoid overflow?

- Do you know any variance reduction technique?
What is control variate method? How and why it works?

- What is the future contract? Under what situation is the future contract priced higher than forward price?
What is the effect of the recent quantitative easing of fed?

- Explain one sorting algorithm. How does the merge sort work? What is the running time for merge sort?
How does the quick sort works? What is the running time for the quick sort?
In practice, how do to choose among these two algorithm

- Given a Brownian motion starting from 0 and two barriers 'a' and '-b', what is the probability of hitting barrier 'a'

before hitting barrier '-b'?
What if it is a process with drift?

- given a set of y_i's and x_i's, what is the linear regression beta? how to interpret in terms of (co)variance? what is the correlation between x_i and the residuals?

- what is the c++ sort function? explain the heap sort. why running time is O(nlgn)? what is quick sort? why it is O(nlgn)

- A european call option with strike 40 expires in 1 year. The current spot price is 50. what happens to the call price as the volatility goes to infinity?

- Given two buckets of volumes 5L and 3L, how to obtain water of 4L?

- A stair with 10 levels. Each time you step up either 1 level or 2 levels. How many different ways are there to get to the top?

- what is AR, GARCH? what is GARCH used for in practice?

- what are the programming languages that you used? how confident are you in programming?

- Describe one of the Monte Carlo simulations you have done.

- Given Brownian motion W, integrate WdW from 0 to T. how to interpret (dW)^2 = dt? how to do the integral by definition? (Shreve's book)

- What is beta given vector observations: Y and X? what is the requirement on X to make it work? what if two observables in X are highly correlated? what technique can be used to fix the problem?

- what is semi-positive matrix? Prove covariance matrix is semi-positive.

- Fibonacci sequence. write a function that recursively calculate F(n). what is the running time? what is the asymptotic limit of F(n)? How to obtain simple lower and upper bound of F(n) when n is large.

- Give an situation when you could have a memory leak.

- Randomly pick three points, what is the probability that they will be on the same hemisphere?

- How to price European option using Monte Carlo? (Assuming stock price follows geometric brownian motion) Is Monte Carlo applicable in pricing American option?
How does the binomial tree pricing work for American option?

- Place cubes on a 10x10 grid. Place 1, 2 ,3 ..., 10 cubes in the first line. 2,3,4,...,11 cubes in the second line. 3,4,5,...,12 cubes in the third line. ... 10,11,12,...,19 cubes in the 10th line. what is the total number of cubes(intuitively)?

- X1, X2 are standard normal, with correlation rho. What is the variance of X=X1+X2.

- X1 and X2 are independent random variable with pdf f and g. what is the pdf of X=X1+X2? What is the relation between Fourier coefficient of convolution of two functions and that of individual functions?

- Explain the copy constructor.

- Given uniform(0,1), how to generate a random variable with pdf f ?

- Given an example of variance reduction technique.

- What is the distribution of independent standard normals?

- Given two arrays, each with distinct integers, give an efficient algorithm to find out the common elements.

- How to draw normal random given uniform variable generator?

- Show that geometric Brownian motion leads to log normal.

- A survey claims that 70% of people like coffee, 80% of people like tea, what is the upper and lower bound on the percentage of people like both coffee and tea?

- On a highway, the probability of observing at least one car in 20min is 609/625, what is the probability of observing at least one car in 5min?

- Derive the Black-Scholes differential equation.

- Random walk starts from 0. what is the probability of hitting -2 before hitting 3?

- Toss a dice. What is the expected number of tosses before getting two consecutive 6's?

- Given a simple example of shell script that repeat certain task 10 times.

- What is a Brownian motion? What is the correlation of W(t1) and W(t2)?

- Algorithm to reverse the characters in a char array. Algorithm to reverse the word order in a sentence with words seperated by white space.

- 100 bulbs initially all off. First switch (on <-->off) the states of 1,2,...,100'th bulbs. The second time switch the states of 2,4,6,...,100'th bulbs. (the i'th time switch the states of i,2i,3i,... bulbs). The last time switch the states of 100'th bulb. What bulbs are on in the end?

- Asked about resume 100 times. Explained my research 100 times.

- Describe your thesis/research.

- Describe class hierarchy of any large computing system that I have experience on.

- What is the "volatile" keyword?

- What is the "static" keyword?

- If one tosses a fair coin 5 times, what is the probability of not getting 2 heads or 2 tails in a row?

- The ransom note problem. Design an algorithm that makes up the ransom note from characters in a magazine.

- Introduce yourself

- What is a virtual function?

- Describe any sort algorithm

- How to derive Black-Scholes equation

前一阵微软面试两次。第一次on-site fail, 第二次过了。On-site照例先HR谈话，说我们今天只安排3轮，别以为到中午走人就是fail等等假话套话；3轮过后，如果过不了就走人。留下来的lunch interview, 吃完饭和同一个面试官再谈20～30分钟，然后再来一个最高的manager面试。下面是面试技术题目的总结。其他问题包括介绍一下自己，怎么处理同事间矛盾，对我们公司会有什么贡献，等等，都是套路。

电话面试

1) heap/stack的概念，区别？
2) virtual function? pure virtual function? virtual desctruction? 概念？何时用？
3) Base class 和derived class之间的赋值(比如A是base, B是derived, A p1 = new B, 行不行？反过来呢？)
4) class的概念，为什么要用OOP? protect/public/private，什么时候用它们？
5) 设计一个parking lot的class
6) 线程之间怎么通讯？什么是critical section/semaphore/mutax，区别？
7) process和thread的区别？

8) 怎样检测link-list中的circle?

```
#include<iostream>
using namespace std;
struct ListNode
{
   int data;
   ListNode *next;
};
class LinkList
{
   ListNode *head;    //指向链表的头指针
   int count;         //结点数目
   ListNode *interP;  //存放制造环时的中间节点
public:
   LinkList();
   LinkList(int n);    //手动创建一个含有n个元素的链表
   ~LinkList();
   void ReverseList();    //逆序链表
   void PrintList();      //打印当前的链表
   bool IsCircle();       //判断链表中是否有环
   //制造环——方法是将链尾指向链中第n个节点后面(条件:count>n)
   void GeneCircle(int n);
   void DegeneCircle(int n);  //拆环，防止不能析构
};
LinkList::LinkList()
{
   head = NULL;
   count = 0;
}
LinkList::LinkList(int n)
{
   count = n;
   head = new ListNode;
   ListNode *p = head;
   for(int i=0;i<n;i++)
   {
      ListNode *newPtr = new ListNode; //新建一个结点
      cout<<"Input Node "<<i+1<<": ";
      cin>>newPtr->data;             //新节点赋值
      p->next = newPtr;              //当前结点指向newPtr
      p = p->next;                   //p变为当前位置的下一个位置
   }
   p->next =NULL;
   interP = NULL;
```

```
}
LinkList::~LinkList()
{
   ListNode *p = head->next;
   ListNode *tmpPtr;
   while(p != NULL)
   {
      tmpPtr = p->next;
      delete p;
      p = tmpPtr;
   }
   delete tmpPtr,p,head,interP;
   p = tmpPtr = head = interP = NULL;
   cout<<"Deconstructor!"<<endl;
}
void LinkList::ReverseList()
{
   ListNode *pCurr = head->next;  //pCurr指向待逆序链表的表头结点(即p指向第一个节点)
   if(pCurr == NULL || pCurr->next == NULL)  //只有头指针或者只有一个节点
      return ;
   head->next = NULL;           //清空原链表(head仍为逆序后的表头结点)
   //将原链表中的结点依次进行逆序链接
   while(pCurr != NULL)
   {
      ListNode *pTmp = pCurr->next;  //pCurr指向待处理的结点,pTmp存放pCurr的下一个节点
      //将pCurr插入到head和第一个节点之间(即将pCurr作为新的第一个结点)
      pCurr->next = head->next;     //将新的待处理的节点pCurr插入到原先第一个节点(head->next)之前
      head->next = pCurr;          //头结点指向新的节点pCurr
      pCurr = pTmp;              //下一个待处理结点
   }
}
void LinkList::PrintList()
{
   if(count == 0)
   {
      cout<<"List is empty!"<<endl;
      return;
   }
   ListNode *p = head->next;
   cout<<"data of the List is:"<<endl;
   while(p != NULL)
   {
      if(p->next)
         cout<<p->data<<"-->";
      else
         cout<<p->data;
      p = p->next;
   }
   cout<<endl<<"length of the list is: "<<count<<endl;
}
bool LinkList::IsCircle()
{
   ListNode *p = head->next;
   ListNode *q = head->next;
   while(p->next && q->next)
   {
      p = p->next;
      if(NULL == (q = q->next->next))
         return false;
      if(p == q)
         return true;
   }
```

```
    return false;
}
void LinkList::GeneCircle(int n)
{
    if(n >= count)
    {
        cout<<"can not generate a circle!"<<endl;
        return;
    }
    ListNode *temp,*p = head->next;     //p将要指向第n个（队尾）元素
    for(int i=1;p->next;i++)
    {
        if(i == n)      //找到第n个元素
        {
            temp = p;   //先用temp暂存第n个元素的指针p
            break;
        }
        p = p->next;    //p不断指向后面的元素
    }
    interP = p->next;
    p->next = temp;     //p的下一个元素指向temp（暂存原先p指向的元素），这样就形成了一个环
}
void LinkList::DegeneCircle(int n)
{
    ListNode *p = head->next;
    for(int i=1;p->next;i++)
    {
        if(i == n)
            break;
        p = p->next;
    }
    p->next = interP;
}
int main()
{
    int N,pos;
    cout<<"Input length of the List: ";
    cin>>N;
    LinkList ls(N);
    ls.PrintList();
    ls.ReverseList();
    ls.PrintList();
    ls.ReverseList();
    ls.PrintList();
    cout<<"Input the position where to generator circle:";
    cin>>pos;
    ls.GeneCircle(pos);
    if(ls.IsCircle())
    {
        cout<<"Circle Linklist!"<<endl;
        //必须使用这个函数恢复链为无环状态，否则因为环的存在，
        //链表的析构函数不能正常退出，导致程序死循环
        ls.DegeneCircle(pos);
    }
    else
        cout<<"Non-Circle LinkList!"<<endl;
    return 0;
}
```

9)　写一个reverse string的代码，怎么测试？(网上用livemeeting的面试)

```
void reverse_str(char *ch)  /*使用中间变量*/
```

```
   {
     int len;
     int i;
     len = strlen(ch)-1;
     char ctemp;

     for(i = 0; i < len-i; i++)
     {
          ctemp = ch[i];
          ch[i] = ch[len-i];
          ch[len-i] = ctemp;
     }
     ch[len+1] = 0;

   }

-------------------------------------------------------

   void reverse_str2(char *ch)   /*不用中间变量*/
   {
     int len;
     int i;
     len = strlen(ch)-1;
     char ctemp;

     for(i = 0; i < len-i; i++)
     {
          ch[i] = ch[i] ^ ch[len-i];
          ch[len-i] = ch[i] ^ ch[len-i];
          ch[i] = ch[i] ^ ch[len-i];
     }
     ch[len+1] = 0;
   }
```

10)　写代码，计算一段plain text中有多少单词，以及计算出现频率最高的单词(网上用livemeeting的面试)。怎么测试？

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
typedef struct Word {
 char w[20];
 int  k;
 struct Word *next;
}pWord;

int main(int argc, char *argv[])
{
 FILE *fp = fopen("input.txt","r");
 struct Word *Head = NULL;

 while (!feof(fp)) {
  char *p = (char *)malloc(20*sizeof(char));
  fscanf(fp, "%s", p);
  if(Head == NULL){
   struct Word *temp = (struct Word *)malloc(sizeof(struct Word));
   strcpy(temp->w, p);
   temp->k = 1;
   temp->next = NULL;
   Head = temp;
  } else {
   struct Word *pp = Head;
   while (pp != NULL) {
    if (strcasecmp( pp->w, p) == 0){
     ++pp->k;
     break;
    } else {
     pp = pp->next;
    }
   }
   if (pp == NULL){
    struct Word *temp = (struct Word *)malloc(sizeof(struct Word));
    strcpy(temp->w, p);
    temp->k = 1;
    temp->next = Head;
    Head = temp;
   }
  }
 }
 struct Word *q = Head;
 while (q != NULL) {
  printf("%s    ", q->w);
  printf("%d\n", q->k);
  q = q->next;
 }
 return 0;
}   /*------end of main------*
```

11) A. Write a program in C which takes a string and tests whether any permutation of its letters could be a palindrome. Optimize for speed. Make sure your code handles all error conditions. Example : if input string is " LIILR" then your program should output TRUE. Since one of its permutations is LIRIL which is a palindrome. B. How would you change the above program if you also had to print out the palindrome. Example : if input string is "LIILR"

then your program should output TRUE, LIRIL., 限时45min, email寄回。

12) Deadline邻近，程序出现重大问题，要么是没时间修改，要么是程序员拒绝修改，怎么办？

13) Deadline邻近，没有时间做足够测试，怎么办？如果连排在priority list中的项目都没时间测试，怎么办？

14) boost pool memory概念

15) java monitor概念

Onsite (基本所有代码都要说出复杂度):

1) 给一段代码，找错误

2) 什么是deadlock, 怎么避免？

进程无限期等待不可能得到的资源，就是死锁。

2.死锁的防止（要求达到“简单应用”层次）。

（1）系统出现死锁必然同时保持的四个必要条件。

（2）死锁的防止策略：静态分配、按序分配、抢夺式分配。

3.死锁的避免（要求达到“简单应用”层次）。

（1）安全状态。

（2）区分死锁的避免与死锁的防止。

（3）银行算法是怎样避免死锁的。

4.死锁的检测（要求达到“领会”层次）。

（1）什么是死锁的检测。

（2）怎样实现死锁的检测。

（3）检测到死锁后的恢复工作。

3) 代码：Reverse一个link-list， 怎么测试？

```
class Node
{
public:
        Node(int value) : value(value), next(NULL) {}
public:
        int value;
        Node* next;
};
Node* reverseList(Node* head)
{
        Node* newList = NULL;
        Node* current = head;
        while (current)
        {
                Node* next = current->next;
                current->next = newList;
                newList = current;
                current = next;
        }
        return newList;
}
```

4) 代码：link-list按照value合并/排序，怎么测试？

```
#include <stdio.h>
#include <stdlib.h>

struct node {
 int number;
 struct node *next;
};
```

```
/* add a node to the linked list */
struct node *addnode(int number, struct node *next);
/* preform merge sort on the linked list */
struct node *mergesort(struct node *head);
/* merge the lists.. */
struct node *merge(struct node *head_one, struct node *head_two);

int main(void) {
 struct node *head;
 struct node *current;
 struct node *next;
 int test[] = {8, 3, 2, 6, 1, 5, 4, 7, 9, 0};
 int i;

 head = NULL;
 /* insert some numbers into the linked list */
 for(i = 0; i < 10; i++)
  head = addnode(test[i], head);

 /* sort the list */
 head = mergesort(head);

 /* print the list */
 printf(" before  after\n"), i = 0;
 for(current = head; current != NULL; current = current->next)
  printf("%4d\t%4d\n", test[i++], current->number);

 /* free the list */
 for(current = head; current != NULL; current = next)
  next = current->next, free(current);

 /* done... */
 return 0;
}

/* add a node to the linked list */
struct node *addnode(int number, struct node *next) {
 struct node *tnode;

 tnode = (struct node*)malloc(sizeof(*tnode));

 if(tnode != NULL) {
  tnode->number = number;
  tnode->next = next;
 }

 return tnode;
}

/* preform merge sort on the linked list */
struct node *mergesort(struct node *head) {
 struct node *head_one;
 struct node *head_two;

 if((head == NULL) || (head->next == NULL))
  return head;

 head_one = head;
 head_two = head->next;
 while((head_two != NULL) && (head_two->next != NULL)) {
  head = head->next;
  head_two = head->next->next;
 }
 head_two = head->next;
 head->next = NULL;

 return merge(mergesort(head_one), mergesort(head_two));
}

/* merge the lists.. */
struct node *merge(struct node *head_one, struct node *head_two) {
 struct node *head_three;

 if(head_one == NULL)
  return head_two;

 if(head_two == NULL)
  return head_one;

 if(head_one->number < head_two->number) {
  head_three = head_one;
  head_three->next = merge(head_one->next, head_two);
 } else {
  head_three = head_two;
  head_three->next = merge(head_one, head_two->next);
 }

 return head_three;
}
```

5) 代码：设计一个vector(矢量)的class

6) 代码：reverse一段明文中的各个单词, 怎么测试？

7) 代码：程序接受command line input, 比如说程序是mycalc.exe, 用户运行”
mycalc 3+2*3*3-1”，输出答案。假设: 1. 没有除法； 2. 用户输入正确(不会出现“
mycalc 4+-5”类似情况)；3. 只考虑integer。 代码写完当场输入计算机测试

8) 4x4的正方形，有一些空格已经填入数字(1~4)，条件：每行每列以及每个2x2的
小矩形(只考虑边缘4个小矩形)，1~4四个数字只能出现1次。设计算法。
9) 有个party, 请了个人过来演讲，这个人不认识听众的任何一人，任何听众都认
识他。现在演讲者已经混入人群，你可以问任何一人”你认识哪些人？”(当然不能问“
谁是演讲者”)，优化算法，最快找到这个演讲者
10) 写代码的时候，往往会给array定一个max_number, 如果现实中有可能出现高于
这个数字，怎么测试？比如找出邻近的wifi网络数，max_number再怎么大，总有可能超出
，怎么测试呢？而且比如这种情况还很难模拟(很难在现场设几百个网络吧)，怎么测试？
不准用动态数组。

google interview question from glassdoor

Design and describe a system/application that will most efficiently produce
a report of the top 1 million Google search requests. You are given:

You are given 12 servers to work with. They are all dual-processor machines
with 4Gb of RAM, 4x400GB hard drives and networked together.(Basically,
nothing more than high-end PC's)
The log data has already been cleaned for you. It consists of 100 Billion
log lines, broken down into 12 320 GB files of 40-byte search terms per line.

You can use only custom written applications or available free open-source software.

题目：

从一个string 变到另一个，比如"study"->"world" （字数相等），要求

1. 每次变一个字母
2. 每次改变后的string必须是一个词典里面能查到的英语单词，比如你不能把study变成atudy

编程题
given a character string, print the number of occurence of each charcater in order. ie. if the string is "ceabcw", then you should print something like:
a 1 b 1 c2 e 1 w 1.
she asked the possible data strucutre to approach. I gave array, hashtable, and BST. she asked me to use BST, and using no recursive. Also how to handle unicode.
然后问了一些测试题，让我测试她们的一个产品。细节忘了，总之她对我不满意。我也觉得基本没戏了。
第二个是个印度人。编程题：
given a matrix(assume it is a bitmap), print all cells that is on.
做的不好。后来问了一些测试题。
第三个是个白人。
开始问测试的问题，回答得一般。因为觉得已经没有戏了，所以也不大有精神。
编程题很简单，是实现阶乘。不过有个问题没有考虑到，就是overflow怎么处理

Given a document of text, return the 10 most frequently occurred words in the text.

比如说叫design一个parking lot, elevator，online shopping system.

这些问题应该怎么approach,主要是考察什么方面的知识和能力呢？ OOP和design pattern?那么是不是说，一定要考虑到composition,还是inheritance,各自的优缺点要说下？ 尽量避免coupling之类的?要不要说个design pattern之类的，比如，singleton,或者factory.

Q2) Given a list of characters and an int which is the distance between the same characters

Eg: input- aaaaabbbbcc and distance as 2. One of the outputs can be- aabaababbcc

Come up with an algorithm and Code it.

好吧，我承认，给我的人说他也不记得了，当时就是挺confused的。

来另一个吧：

Q2) Find out if two inputs are Anagrams with HUGE HUGE input (like thousand of terabyte)

Q3) Given lots and lots of points in a 2D space find all the line with most points on it.

很简单，就是check 一个array里面有没有两个元素和为sum,

我用了两种方法，一个是sort之后，keep两个指针i,j一头一尾,相加如果小于sum就前移，否则就后移。O(nlgn) time

另一种方法是hash table.其中一个trick就是要确信没有自己加自己。这个是唯一一个
主意点吧。O(n) with O(n) space

1. sort stack using only pop(), top(), push(), isEmpty(),isFull(). Do not use any auxiliary stacks or arrays.
   感觉不用辅助空间做不到啊，这题是真的有很巧妙的方法？还是玩玩文字游戏--比如用linked list然后宣称它不是stack也不是array...

   2. Given a set of coin denominators, find the minimum number of coins to give a certain amount of change
   贪心和DP,回溯法应该都能做的，无非就是状态空间搜索，但是我想到的都是伪多项式的算法，感觉这道题应该能利用某些数论的知识在多项式时间解决

. 两个C的程序问题
先是char*指针问题

```
char *dosth()
{
char s[256];
char * p = r;
p = "some new string":
}
```

然后问了一堆变量的值，比如 s, *s, *(s+2), &p, etc.

另外一个switch程序找错，没有加break之类，还有就是return local variable地址的
问题

3. 手写fab(n)函数，不是算，而是输出，递归或者循环都可，不过递归不高效大家应
该知道

4. 逻辑问题：八个水罐称重

5.
一堆关于OO概念的问题，多态，继承，封装，接口和抽象类的区别，复写和重载（包括C++具体怎么实现的
）

6. 反馈问题

第二波一个项目经理
一来就是比较高难度的，给你一个字节数组（注意取值范围），数组长度可能非常长，
如何找到第一个只出现了一次的数字。开始没什么思路，和他讨论了一会，边问还边问
复杂度和数据结构的问题，后来发现应该进行数出现次数，这样复杂度就是２ｎ，结果
出来了要求手写出代码。

然后就是一个智力问题，三个囚犯黑帽白帽.

实常见，好象没见过好解。
Write a program to evaluate a simple mathematical expression like 4 + 2*a/b
– 3
基本idea就是要用stack来记录原来的operand, 然后新的operand和在stack上的
operand进行比较。可能要计算中间值（一旦precedence确定），然后再push 进stack.
但是具体怎么操作觉得没想明白，哪位大写指点一下？？多谢！！

1. Given a set of n points, find the line that intersects the most number of points
   想不出有比n^2更好的办法

   2。Given n integers, find two of them that has max xor result

如果假定a xor b takes time len(a)+len(b)), then the naive way will take time n^2 * avglen
my idea: first align all the numbers to same length, say it is x, then distribute by most significant bit, as 1, 0, to two array one and zero. we will keep dividing one into oneone, onezero, zero into zerozero, zeroone. then to achieve the max xor, we must pair oneone with zerozero, zeroone with onezero if possible. and keep doing this. the expected time should be nlgn* x if groups of words with prefix 00, 11, 01, 10 have the same size.

Design a system to store heap on multiple machines ? What is avg number of machines accessed per operation and number of elements stored in a machine ?

First greater number in an array. Given a large array of positive integers, for an arbitrary integer A, we want to know the first integer in the array which is greater than or equal A . O(logn) solution required
ex [2, 10,5,6,80]
input : 6 output : 10
input :20 output : 80

Given an N-by-N array of black (1) and white (0) pixels, find the largest contiguous sub-array that consists of entirely black pixels. In the example below there is a 6-by-2 sub-array.

```
1 0 1 1 1 0 0 0
0 0 0 1 0 1 0 0
0 0 1 1 1 0 0 0
0 0 1 1 1 0 1 0
0 0 1 1 1 1 1 1
0 1 0 1 1 1 1 0
0 1 0 1 1 1 1 0
0 0 0 1 1 1 1 0
```

1. 很长的log file记录了用户访问amazon.com的过程，两列分别为 userID 和 pageName.
log从上倒下按照点击发生的时间顺序。找出最popular的3连击。
eg:

```
zhang  welcome
Li     Hello
Wang   welcome
Li     books
Wang   Hello
zhang  books
Li     shopping cart
Li     checkout
zhang  shopping cart
Wang   camera
zhang  checkout
```

最popular的3 combo是books -> shopping cart -> checkout

2. Permutation of a string.这题最郁闷，我把programming expose里的code默写了出来。但这个方法是不管字符重复的，假设都是不同的。现在考官要不显示重复的，而且他要求不能先
都列出来再剔除，而要在发现重复的时候及时制止。没想出来

3. Design a fight ticket booking system.

4. 老板说网站很慢怎么办？
老板说数据库很慢怎么办？

1 什么是标准方差

2 什么是标准error
3 operator new什么功能
4 virtual function干什么用用的, 为什么要有virtual function
5 dynamic binding如何实现的
6 如何debug 程序, 如果是多线程的如何debug
7 malloc和calloc的区别
8 什么是abstraction, 如何实现abstraction
9 写一个程序计算fabonaci, 如果不recursive怎么写, 如何继续优化,我最后给的答案是用两个变量就行了.

2. phone interview with 2 people in R&D, Dec 2009
   1) how to find 1 missing number from 0 to N in an array of N numbers.
   2) brainteaser, 5 jar problems.
   3) how to calculate sqrt(N) without using sqrt function.

```
#include <iostream>
#include <math.h>
```

3.

```
using namespace std;
```

4.

5.

```
float sqroot(float m)
{
```

6.

```
    float i=0;
   float x1,x2;
```

7.

```
   while( (i*i) <= m )
          i+=0.1;
```

8.

```
   x1=i;
   for(int j=0;j<10;j++)
```

9.

```
   {
        x2=m;
```

10.

```
        x2/=x1;
        x2+=x1;
```

11.

```
        x2/=2;
        x1=x2;
```

12.

```
    }
    return x2;
```

13.

```
}

```

14.

```
int main()
{
```

15.

```
    cout<<"Enter a Number:";
    int no;
```

16.

```
    cin>>no;
      cout<<"Square Root using sqroot()= "<<sqroot(no)<<endl
```

17.

```
         <<"Square Root using sqrt()  = "<<sqrt(no);

```

18.

```
    return 0;
}
```

19.

20. Binary search tree problem.

3. onsite interview, Jan 2010
1st meet 2 people in R&D
1) train, tunnel, people escaping problem
2) 6 digits number, each changes from 0 to 9. Find the odds that sum of first three is the same as the sum of last three. A: 2 do loop.
3) Find 1 missing number from 0 to N. But notice that it is possible the sum would overflow. Think about a way to avoid the overflow.

4) Tricky problem. I do not think anyone else would know the answer except the one who gives the problem. Nothing to do with math, statistics.
5) Same 5 jars problem. That is their favorite.

2nd meet a lady in HR.
Ask 15-20 Behavioral problems. Cover most commonly behavioral problems.

3rd meet a senior manager in R&D
Talk a lot about my research.
Ask one question, how to find the first unique number in an array of byte.
and write a code to realize it.

题目1. LIS. 一个任意的数组，找出一个严格单调递增的最长子序列。
例如: {3,0,1,7,2,4,5,9} –> output： {0, 1, 2, 4, 5, 9}
很简洁巧妙的算法，能在O(N log N)时间和O(N)空间做出来！方法就是始终保持一个单增的序列，然后新来的数如果比当前最大还大就append在后面，否则在单增序列里面做binary search，替换相应位置的数。

题目2. 玻璃杯/鸡蛋drop问题。有N层楼，假定是在 i 层楼扔鸡蛋，如果没有碎，那么在所有<=i 楼层扔鸡蛋都保证不会碎，反之如果碎了，那么保证在所有 >=i 楼层扔鸡蛋都必碎。通过若干次尝试扔鸡蛋，找到某个鸡蛋碎/不碎的”临界”层。允许你扔鸡蛋的总次数是D，允许你打碎的鸡蛋数是B。

问题的描述是：对一组给定的数（N D B），如果存在一个策略保证能在D B的限制下，在N层楼中找到“临界”层，那么称此(N D B)是Solvable的。接下来相关联的三个问题就是:
（a）给定D，B，求满足（N，D，B）Solvable的最大的N_max. 例：D=4，B=1， 策略是从第一层开始一层层往上. N_max=D=4.
（b）给定F，B，求最小的D_min
（c）给定F，D，求最小的B_min
这个问题相当容易找到看似最优的解，但是绝大部分的方法都不是最优的(最快最高效)。而且最迷惑人的是，(a)(b)(c)三个问题中，必须先从其中某一个下手开始解决，如果你不幸的先从另外的两个问题下手，多半离最优解遥遥无望。

如果你找到了正确的入手点，有了正确的思路，最后的答案会异常的简单！
入手点就是首先解决(a)问题，并且可以递归的来解决：假设D,B对应的答案是F(D,B)，那么考虑在某一层摔一个鸡蛋后，如果碎了，D--,B--，如果没碎就只是D--，B不变。这样很容易写出递归方程，算出F关于D,B的table。

题目3. 经典的概率悖论。3扇门，一扇背后有羊，你选中一扇门后，现在另外一扇门开了，里面是空的。问你是否应该重新选择。

分析：据观察，有一部分的人坚持认为一定要重新选择，另一部分的人认为是否重新选择都一样。另外少部分的人能看出，这个问题很巧妙的隐含了意识(主观intention)，信息和概率的关系！

题目4. 很简单的，N个数的数组，找出最大的和第二大的数，只用N+logN-2的比较次数，不需要额外空间。这个是典型的问题本身就是答案提示的题目--基于比较又有LogN，很显然思路涉及二分法，继续下去，剩下的问题就仅仅是找一个符合要求的Implementation了。

题目5. 找N！最后一个非零的数字。巧妙的方法可以在 LogN 时间内找出来，一个hint是利用 5^k（和log_5）划分问题
题目6. 任务分配，假设有N个任务，每个任务需要W_i工作量，M个人，每人每天能做工作量w_i，如何安排工作，使得所有工作能最快完成。这个问题其实更像一个开放性问题，因为一个合理的贪心策略，最后的结果跟最优结是很接近的(大致上，最多只差一天)。

题目7. 计算Fibonacci 数 F(n)，O(n)的算法是很trival的。但是有很漂亮简洁的Log(N)算法，思路是利用2*2矩阵表示Fibonacci递推式，然后用二分法的思想球矩阵的N次方。

题目8. 一颗BinaryTree，每个节点有个NULL指针，要求把每个节点和在BFS中它的下一个节点串起来。其他BinaryTree的常见题有比如非递归的实现遍历，用.parent or stack。思考这些题的经验是，对于这一类的树的题目，有很强的递归性/规律性，通常都是O(N)的复杂度，那么把N steps的问题，放在某个单step来研究，会把思路变得更清晰。另外一点就是，完全可以假设在做这一单步之前，在做这一步之前的问题已经最大可能的正确解决了，这样能够以一种数学归纳法的思想去利用之前的结论。比如这个题里面，假设节点 i 之前的节点都已经串好了，如何把 i 串到下一个节点。这个问题就是看一眼草图就能知道的了。最后一点经验是，在效率相当的算法的基础上，不同版本的实现，已经有能够互相启发的地方。

//

Given two arrays A [1..n] and B[1..m], find the smallest window in A that contains all elements of B. That is, find a pair <l,k> such that A[l..k] contains B[1..m]

For example, given A = 3,1,5,7,3,5,2 and B = 5,3 then the smallest window is [3,5].

. 一个sorted interger Array[1...N], 已知范围 1...N+1. 已知一个数字missing。
  找该数字。
  把原题改为unsorted，找missing数字。 performance。

2. 复制linked list。 已知每个节点有两个pointer，一个指向后一个节点，另一个指向
  其他任意一节点。 O(n)时间内，无附加内存，复制该linked list。（存储不连续）

3. 一个party N个人，如果一个人不认识任何其他人，又被任何其他人认识，此人为
  celebrity。用O(n)时间找到此celebrity。

4. 给中序后续，构建树。

Brain teaser:
1. 50个黑球50个百球，2个罐，要求你放这100个球在这2个罐，使得别人随机从2个罐中任意拿一个球是黑球的几率达到最大。

2. heard on the street 上的男人出轨题，简单逻辑推理。

3. 这个没答上来，后来给了提示做出来了，但是回头想想还是不对。上来请教一下。

2个人商量好策略，然后一个从52张牌里面随机抽5张，看牌，考虑。。。然后排在桌上，摊开前4张，第5张面朝下，由第二个人判断第5张牌。 问这个策略。

. 写atoi函数，

```
// Test2.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <cassert>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <vld.h>

using namespace std;

double Strtod(char *str, char **endstr)
{
    double num1 = 0.0;
    double num2 = 0.0;
    double point = 0.1;
    int sign = 1;

    int len = strlen(str) + 1;
    *endstr = new char[len];
    memset(*endstr, 0, len);

    if (*str == '-')
    {
        sign = -1;
        ++str;
    }

    if (!isdigit(*str))
    {
        strcpy(*endstr, str);
        return 0.0;
    }
    while (*str && isdigit(*str))
    {
        if (*str == '.')
        {
            ++str;
            while (*str && isdigit(*str))
            {
                num2 += point * ((*str) - '0');
                point *= 0.1;
                ++str;
            }
            strcpy(*endstr, str);
            break;
        }
        else
        {
            num1 = 10 * num1 + *str - '0';
            str++;
            if (!*str || !isdigit(*str))
            {
                strcpy(*endstr, str);
                break;
            }
        }
    }

    return (num1 + num2) * sign;
}


int  Atoi(const char *pstr)
{
    int sign = 1;
    int num = 0;

    while (*pstr == ' ' || *pstr == '\t')
    {
        pstr++;
    }

    if (*pstr == '-')
    {
        sign = -1;
        pstr++;
    }
    while (*pstr)
    {
        if (*pstr >= '0' && *pstr <= '9')
        {
            num = 10 * num + *pstr - '0';
        }
        else
        {
            return num * sign;
        }
        pstr++;
    }
    return (num * sign);
}


double Atof(const char *pstr)
{
    double sign = 1.0;
    double num1 = 0.0;
    double num2 = 0.0;
    double point = 0.1;

    while (*pstr == ' ' || *pstr == '\t')
    {
        pstr++;
    }

    if (*pstr == '-')
    {
        sign = -1;
        pstr++;
    }

    while (*pstr)
    {
        if (*pstr == '.')
        {
            pstr++;
            while (*pstr >= '0' && *pstr <= '9')
            {
                num1 += point * (*pstr - '0');
                point *= 0.1;
                pstr++;
            }
        }
        else if (*pstr >= '0' && *pstr <= '9')
        {
            num2 = num2 * 10 + *pstr -'0';
        }
        else
        {
            return (num1 + num2) * (sign);
        }
        pstr++;
    }
    return (num1 + num2) * (sign);
}

int main( void )
{
    char str[] = "    -1234565kljh";
    int num = Atoi(str);
    cout<< num << endl;

    char pstr[] = "    112adfaf3  43224.569877aa";
    double n = Atof(pstr);
    printf("%6.8f\n", n);

    return 0;
}
```

2. 古老的三角形问题：输入3边，看是什么三角形。
  一个mobile device可以从服务器上传和下载图像，怎么测试这个系统？

3. lunch meeting之后回办公室打开电脑，说他们现在开发的某产品有问题，每次要loading很久,差不多10秒的样子。问怎么测试并找出这个bug？ 这个把我难住了，胡乱讲了一通，然后说太困难了；于是他换了个题目，画了一个plotter软件的界面，问怎么测。

  coding的题目是Path Walk，给一条路径，写一个函数来走通它。其实这个题目我没搞明白什么意思，先沟通了很久，最后开始写（还是不太明白。汗...），写完了觉得不正确，正想再改改，被打住了，说给个test case一起来看看程序怎么执行。每句代码跑了一通，却发现code写正确了:-)

4. 一开始是个IQ题，把一堆数字填到格子里，满足一些条件，比如1和2不能相邻。
  测一个记事薄软件。有scheduler和notifier两部分，可以从scheduler输入时间和内容，然后notifier到预定时间会给出提醒。
  coding题目很容易，找到单链表倒数第N个节点。

5. 最后是hiring manager，问了一些behavior问题，然后打开一个网站，问怎么测试。

在onsite面试中实际遇到的。
1.template中用typename和用class有什么区别？
2.unix下执行shell脚本和执行可执行文件有什么区别？哪个更快，为什么？脚本语言程序（如javascript）和可执行文件程序有什么区别？shell和这两者却别呢？

3.如何对const data member做assignment?

```
class A{
const int a;
public:
    A():a(0){};
    A(int m_a):a(m_a){};
};

int main(){
  A a(1);
  A b;
  b = a; //how to implement assignment for this?
}
```

4.如果把base class对象赋给derived class对象,会怎么样？compiler报错还是执行错误？

```
class A{
public:
  int a;
};

class B:public A{
public:
  int b;
};

int main(){
    A a;
    B b;
    b = a; //what happend?
    cout << b.b << endl;

    B* b2;
    b2 = &a;  //how about this?
    cout << b->b << endl; }
```

2. 两个C的程序问题
先是char*指针问题

```
char *dosth()
{
char s[256];
char * p = r;
p = "some new string":
}
```

然后问了一堆变量的值，比如 s, *s, *(s+2), &p, etc.

另外一个switch程序找错，没有加break之类，还有就是return local variable地址的问题

3. 手写fab(n)函数，不是算，而是输出，递归或者循环都可，不过递归不高效大家应该知道

4. 逻辑问题：八个水罐称重

5. 一堆关于OO概念的问题，多态，继承，封装，接口和抽象类的区别，复写和重载（包括C++具体怎么实现的）

1, C vs C++
2, struct in C v.s. in C++ v.s. class in C++
3, virtual function, pure virtual function, abstract class
   what is the advantages of using virtual function
4, new v.s. malloc()
5, memory for a process (code, static data, stack, heap)
6, how to know the stack is growing in the direction of address increasing or decreasing
7, virtual memory

reverse words in a sentence，使用如下函数。
char* reverseWord(const char* str)

```
#include <iostream>
using namespace std;

void rev(char *l, char *r);


int main(int argc, char *argv[])
{
char buf[] = "the world will go on forever";
char *end, *x, *y;

// Reverse the whole sentence first..
for(end=buf; *end; end++);
rev(buf,end-1);


// Now swap each word within sentence...
x = buf-1;
y = buf;

while(x++ < end)
{
if(*x == '\0' || *x == ' ')
{
rev(y,x-1);
y = x+1;
}
}

// Now print the final string....
printf("%s\n",buf);

return(0);
}


// Function to reverse a string in place...
void rev(char *l,char *r)
{
char t;
while(l < r)
{
t = *l;
*l++ = *r;
*r-- = t;
}
}
```

2.an interger array containing millions of elements with min 0 and max 1000, how to sort it?

```
void csort(int array[], const int end,
      const int max, const int min)
{
  int i;
  const int range = max-min+1;
  int count[range+1],
      scratch[end];

  for(i=0; i<range+1; i++)
    count[i] = 0;

  /* Set the value of count[i] to the number of
   * elements in array with value i+min-1. */
  for(i=0; i<end; i++) {
    int c = array[i]-1-min;
    count[c]++;
  }

  /* Update count[i] to be the number of
   * elements with value less than i+min. */
  for(i=1; i<range; i++)
```

```
    count[i] + = count[i-1];

  /* Copy the elements of array into scratch in
   * stable sorted order. */
  for(i=(end-1); i>=0; i--) {
    int c = array[i]-min;
    int s = count[c];
    scratch[s] = array[i];
    /* Increment count so that the next element
     * with the same value as the current element
     * is placed into its own position in scratch. */
    count[c]++;
  }

  for(i=0; i<end; i++)
    array[i] = scratch[i];
}
```

3.covert interger number to date string, for example, 20090130 -> "01/30/2009"

说说教训：

第一道被输入const给搞死了。先是没有注意const,直接按照常规非const做，没有写完就被叫停了；然后是被平时强调的malloc后必须及时delete规则搞死，坚持认为在函数里malloc一块内存然后在函数外delete是不好的习惯；最后当面试者提出如果定义一块内存，如char tmp[2048]，然后使用会怎么样？自己提到可以在函数外strcpy函数返回结果，却忘了arr大小实际是无法指定的，所以这种方法是不可接受的。总之，很多的trick在里面没有注意到。

第二道使用couting sort应该就可以。面试者要求描述算法，不需写代码。

第三道自己的做法是先取得30,然后01，然后2009然后组合成一个string。问题是这样的话，月和日的01前的0可能会丢失，使得最后结果可能不对。后来想正确的做法应该是先把整形转成string,然后使用substr并组合。另外，面试者问有stl有什么可以替换itoa，自己答不出来。后来查了下，应该是可以使用stringstream来实现，如下：

stringstream ss;
ss << intVal;

ss.str()就是我们要的结果。

为什么对Amazon感兴趣。
* 自己最近的Project。
* 说出自己会的编程语言并打分（1－5）。
* 有没有开发Mobile application的经验。
* 几个常见Data structure的Lookup操作的时间复杂度。
* HTTP post和get的区别。
* **Design Pattern: Singleton, Factory, Lazy initialization。**

```
class Singleton
{
private:
  static Singleton _instance;

  Singleton() {}
  ~Singleton() {}
  Singleton(const Singleton &);             // intentionally undefined
  Singleton & operator=(const Singleton &); // intentionally undefined
public:
  static Singleton &getInstance();
};

// Source file (.cpp)
//
// Static member initialization.
//
Singleton Singleton::_instance;

Singleton &Singleton::getInstance()
{
  return _instance;
}
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
class Computer
 {
 public:
     virtual void Run() = 0;
     virtual void Stop() = 0;
 };
 class Laptop: public Computer
 {
 public:
     virtual void Run(){mHibernating = false;}
     virtual void Stop(){mHibernating = true;}
 private:
     bool mHibernating; // Whether or not the machine is hibernating
 };
 class Desktop: public Computer
 {
 public:
     virtual void Run(){mOn = true;}
     virtual void Stop(){mOn = false;}
 private:
     bool mOn; // Whether or not the machine has been turned on
 };

class ComputerFactory
 {
 public:
     static Computer *NewComputer(const std::string &description)
     {
         if(description == "laptop")
             return new Laptop;
         if(description == "desktop")
             return new Desktop;
         return NULL;
     }
 };

   ////////////////////////////////////////////////////////////////////////////////////////
#include <iostream>
#include <string>
#include <map>

using namespace std;
```

```
class Fruit {
    public:
        static Fruit* getFruit(const string& type);
        static void printCurrentTypes();

    private:
        static map<string,Fruit*> types;
        string type;

        // note: constructor private forcing one to use static
getFruit()
        Fruit(const string& t) : type( t ) {}
};

//definition needed for using any static member variable
map<string,Fruit*> Fruit::types;

/*
 * Lazy Factory method, gets the Fruit instance associated with a
 * certain type. Instantiates new ones as needed.
 * precondition: type. Any string that describes a fruit type, e.g.
"apple"
 * postcondition: The Fruit instance associated with that type.
 */
Fruit* Fruit::getFruit(const string& type) {
    Fruit *f = types[type];   // try to find a pre-existing instance,
or std::map'll create one if not found

    if (!f) {                   // if it was created by map
automatically, it'll be pointing to NULL
        // couldn't find one, so make a new instance
        f = new Fruit(type); // lazy initialization part
        types[type] = f;    // Registering the newly created Fruit in
the types' map for later use.
    }
    return f;
}

/*
 * For example purposes to see pattern in action
 */
void Fruit::printCurrentTypes() {
    if (!types.empty()) {
        cout << "Number of instances made = " << types.size() << endl;
        for (map<string,Fruit*>::iterator iter = types.begin(); iter !=
types.end(); ++iter) {
            cout << (*iter).first << endl;
        }
        cout << endl;
    }
}

int main(void) {
    Fruit::getFruit("Banana");
    Fruit::printCurrentTypes();

    Fruit::getFruit("Apple");
    Fruit::printCurrentTypes();

    // returns pre-existing instance from first
    // time Fruit with "Banana" was created
    Fruit::getFruit("Banana");
    Fruit::printCurrentTypes();

    return 0;
}
```

* Multi-threaded programming, deadlock之类。
* 对Unix环境是否熟悉，几个常见命令，ls, ps之类。
* Reflection的概念，Java reflection，C++里面是不是有reflection。
* 如何实现Garbage Collection。Reference counting的缺点(cycle)，如何解决，JVM
有没有解决。http://hi.baidu.com/abusemind/blog/item/abb6774528141f2ccefca35f.html

* C++里面virtual destructor的用途，于一般virtual function的区别。
* 写一个函数实现两个整数相除，不用"/"和"%"，返回商和余数。写完读给他听。
* 算法设计：一个Galaxy，每个星星用一个三围座标表示，找出离地球最近的1000个

1. How to implement garbage collector (what data structure)
2. How to implement c++ smart pointer
3. Pro and Con of multi process and multi-thread
4. How many stack/heap does a multi-thread program with 10 threads have?

1. 美国人。上来随便聊聊，然后出了个coding 题目 一个数组，找出第一个重复的数
我给了三种方法，最后用hash写的，然后问test case之类的

2.印度人
上来问我会什么C#还是C++，我说C#会的多一些。然后他上来问了四五个简单的 语法
问题。正好我还都会，心理还窃喜着呢。coding 也很简单，给一个01字符串，转化成
整数。写完后 test case。 第二个题目是两个函数互相调用，无限循环了，然我找出
毛病，问怎么解决。
然后午饭跟这个印度人吃，随便聊聊。 就过了

3.欧洲人，不知道哪国。
女的， 人很好，跟她聊的最开心。coding 题目是个没见过的，double bytes
string实现delete键功能。这个比较难解释，她开始也跟我解释了很长时间。 就是删
除字符的时候如何确定是删一个字节还是删两个字节的问题。我给出算法，然后她有提
示有哪些特殊情况要考虑，也做出来了。然后她就问我给一个一般的application 如何
测试，又随便说了一通，结束了

4.美国人 senior test lead
coding 很简单，给一个句子，把里边所有的单词自身reverse
然后给我看他们的产品，问我怎么测试。聊的也挺好

5.欧洲人 director

面到这个人的时候，我都快累趴下了，都不想面了，实在是累。心理还想着，offer拿
不拿得着无所谓，别把老子给累死了。（看来真得努力锻炼身体，不然面试都挺不住）

题目也很简单，找1--100的素数。我就给了最简单的方法，然后我说要 check一些边界
情况，他说不用了。然后让我做到他的椅子上，打开excel，问我怎么测设置字体这个
feature。说完了问我有什么问题没有

给你一本dictionary，任意给你七个letters，让你找出包含这七个字母的、最长的单
词。

条件：可以pre-processing，这样每次给你不同的letters时，可以very effcient

1. REVERSE LINKLIST.
   2.给了N个数，值域[1,N-1]，如何找出第一个重复的数
   3.算POLYNOMIAL，比如5x^4+6x^3-7x^2-8=?
   4.给一个URL,如何把空格这种字符转换成%20这种
   5.给一个LINKLIST,VALUE的指针指向其他NODE，复制他

C++：effective c++上的东西若干；exception相关；继承和子父类指针若干. 十五分钟左右。

算法/编程：1. 大文件随机sample，one pass. 2. sodoku solver. 3. logn解x^y,
4. DP题 5. 1Billion query里选出时间最近5分钟内最frequent的1000个，one pass（我以前在amazon见到过这题）。6.两个排序数组找共同中值。递归和非递归解法。7.斐波那契数列。100层楼梯下楼，可以一步也可以两步，多少种下法？递归和非递归。8 贝叶斯后验概率。9。多少人在一起，生日可能出现重复概率大于0.5？（算法导论原题，我只记得个答案，直接说了。。。）10. 一个数组，找最大值比较次数？同时找最大值和最小值比较次数？找最大值和次最大值比较次数？（他问我是否知道这题，我说是作业题。后来和师兄聊说是这他常拿来用的面试题。）

系统设计和经验：1 设计一个库，提供timer的功能。deltalist/hash，或类似linux kernal的 timer 设计。效率要比较高。2. 一个类似chord的DHT设计。3. 你有一个奇怪的程序，有时有bug，有时没有，说出尽可能多的可能原因。4. printf来debug有何不妥。5. process和thread。process之间的IPC有那些种？process间是否也可以share memory.何时选thread或process。

。C++ 中的virtual destructor是啥？ 为啥要用？
2。quick sort， merge sort的复杂度。
3。 Structure 和class的区别是什么？ （我晕，这个我居然给答反了）
4。关于C++ 处理异常的方法 。 （基本上一头雾水）
5。Monte Carlo method in american style option pricing。 （我说的用least square regression method,blah......）
6. Int_0^T W(t) dW(t) （ 一看见这个，贼激动阿，熟悉的ito' s formula）
7. Stonivich intergral 是啥？ 为什么用Ito's 不用 stonivich? (不知道拼得对不对)
8. 一个国家所有的人如果生了一个男孩以后就停止生育，生了女孩以后就继续生，直到生出男孩才停止生育，问多年以后男孩多还是女孩多？ （要联系上stopping time的概念）。

9。什么是AR model? 啥时候用AR model?
10. American option 的up bound? (我说是stock price,被直接鄙视了， 说更精确的，只好答没有研究过，当时一头雾水)。

1. 用stack class来实现queue，具体用几个stack不限。完了以后问怎么实现thread safety，然后是怎么测试。
   2. 实现strstr(str1, str2)，如果str2是str1的子串，返回true，否则返回false。实现完了以后问如何测试。
   3. 给定一个integer array with both positive and negative numbers，return a contiguous subarray with the largest sum. 我本来想用dynamic programming实现，但面试官希望按照他的一个更heuristic的思路来解，最后勉强搞定。
   4. 给定一个排好序的linked list，删除其中所有的重复元素。比如给定1->2->3->3->4->4->5，返回1->2->5。给定1->1->1->2->3，返回2->3。看起来简单，一边写一边发现许多细节需要小心应对，好在最后搞定。
   5. 给你三个烤箱，每个烤箱可以同时烤两片面包，需要的时间分别是3分钟，4分钟和3分钟。但第三个烤箱有一个slot出了点问题，每次只能烤面包的一面。所以这个烤箱三分钟后只能算烤好一片半面包，你需要把那半片翻个面，在同一个slot里再烤一次才算一片完整的。现在给你这三个烤箱，问烤好21片面包最少需要多少时间？如果是2100片呢？如果是任意给定的N片，要求O(1)时间内给出最少需要的时间。
   6. 给你三根棍子，每根都需要一个小时才烧完，但每根燃烧的速度都不一样，也不均匀。问只有这三根棍子和火柴，如何精确的得到1小时45分钟的计时？

7. 在一个party上，每个人可能认识别人，也可能不认识。现在其中有一个人是名人，定义就是所有的人都认识他，但他不认识其余的任何人。现在要求你去找出这个名人来。但你只可以通过一个方法，就是问A是不是认识B，回答是表示A认识B，不是表示A不认识B。你可以任意去问这样的问题，问最少需要多少次能找出这个名人？思路有了之后要求写代码实现，可以调用knows(A, B)，代表上面的那个问题。实现完了以后问如何测试
8. 测试copy这个命令。然后自己问了一些clarifying questions，搞清了实际上是copy src dest。src可以是文件，也可以是目录。dest可以存在，也可以不存在。

OO设计题，怎么做一个十字路口的traffic light.
2. 怎么不用recursion 做二叉树in order 遍历

1. Write a function that returns a node in a tree given two parameters: pointer to the root node and the in order traversal number of the node we want to return. The only information stored in the tree is the number of children for each node.
2. 2. Input a message and a text, find if the message can be composed by the text.
   If the text is in a magazine (two pages/a paper), how to design an algorithm

1When casting an object of a polymorphic class from a base calss type, which one of the following casts
performs the task only if the cast is valid?

a. static_cast
b. (void*)
c. dynamic_cast
d. const_cast
e. reinterpret_cast

```
2.  class A
{

public:  void f();
protected: A() {}
      A(const A&) {}
};
```

why are the default and copy constructors declared as protected?

a. to ensure that instance of A can not be created via new by a more derived class
b. to ensure that instance of A can only be created by subclasses of A
c. to ensure that isntance of A can not be copied
d. to ensure that A cannot be used as a base class.
e. to ensure that A cannot be instantiated on the stack

```
3. template<class T1; class T2; class T3>
int Product(T1 a, T2 b, T3 c)
{
   return a*b*c;
}
```

what is wrong with the sample code above?

a. templates must be class definitions
b. the template parameters should be separated by commas.
c. the template definition is missing a pair of braces.
d. template parameters must be pointer types.

e. the * operator has not been defined for T1, T2, and T3.

```
4. class FOO
{
   char * buf;

public:

   Foo (const char *b = "default")
   {
      if (b)
       { buf = new char[std::strlen(b) + 1];
         std::strcpy(buf, b);
       }

      else

         buf=0;

      }

   ~Foo() { delete[] buf; }
};


Foo func (Foo f)
{
   return f;
}
```

when the function fun is called, the program may crash or exhibit unexpected behavior, what is the reason ofr this problem?

a. the destructor may attempt to delete the string literal "default"
b. the destructor needs to check that the value of buf is not 0.
c. the class does not allocate a long enough buffer.
d. the function needs to return Foo& instead of Foo.
e the class needs to specify a copy constructor and assignment operator.

1．请书写一个程序，将整型变量 x 中数字左右翻转后存到另外一个整型变量 y中，例如 x = 12345 时，y为 54321，x = －123 时，y为－321。其中 x 的个位不为 0。

```
void reverse (int x, int* y);
```

(1) 请实现该函数，以上函数原型是用 C语言写的，你可以用你熟悉的语言；
(2) 请写出一段代码验证该函数在各种情况下的正确性。

2．对集合{1, 2, 3, …, n}中的数进行全排列，可以得到 n!个不同的排列方式。现在我们用字母序把它们列出来，并一一标上序号，如当 n=3 时：

```
0．123
1．132
2．213
3．231
4．312
5．321
```

现在，请书写一个函数 void print (int n, int k)，（函数原型是用 C语言写的，你可以用你熟悉的语言）在已知 n和序号 k 的情况下，输出对应的排列，并简要阐述思路。

3．一维数轴上有 n 条线段，它们的端点都是已知的。请设计一个算法，计算出这些线段的并集在数轴上所覆盖的长度，并分析时间复杂度。例如，线段 A 的坐标为[4, 8]，线段 B 的坐标为[1, 5.1]， 那么它们共同覆盖的长度为 7。 请尽量找出最优化的算法， 解释算法即可，不必写代码。

How do you know the bloomberg?
What position do you expect?
What language do you want to answer with? (I choose C.)
What kind of questions do you meet for the online assessment?

what is static in C? how is it implemented by the compiler?
write the definition of a function that returns both the max and min.
why do you use the condition variable?
how to implement a lock?
Under what condition must you use linked list instead of array?
what data structure can you use to store elements dynamically and access them efficiently?
The complexity of finding any element in a linked list in the worst case.
multi-thread library programming: did you write your multi-thread library with p-thread? is there any problem you have with you library?
did you do your projects on linux? If you want to find a string in a file, what command should you use?
do you know vector in C++?

给定 X[1..n] and Y[1..m] 两个 arrays，已经sort好了.
如何找到X <Union> Y的median？我说用merge sort，要O(m+n/2)时间。面试官明显不满意。
这个已经 lineal了？难道还有更快的？

这都是amazon的题目吧
1.sum of all nodes in a path = givenValue
2.http://www.careercup.com/question?id=87897

是现场写code的面试。
第一道是写一个函数，两个参数（String prefix, String s）, 返回true如果s有prefix
第二道是写一个函数，两个参数（int[] a, int sum）, 找出数组里加起来是sum的几个数

1. Coding题非常old了。两个string找最长common substring。这个当场肯定code不了subffix tree。于是就用暴力的方法，三下五除二搞定。然后问complexity，如何改进，bla bla bla。竟然忘记了说可以用DP，低级失误啊。不过面试官还算满意。

这道题被好几个不同的公司面到过：Fibonacci数列，一般让你给一个recursive的版本，然后写个iterative的版本，然后问有没有更快的可能性。我记得以前在某个版讨论过，参考wiki:

这样的方法，可以在O(log(N))的时间和O(1)的空间复杂度内算好。要写程序的话，用类似下面的方法：

```
Matrix2x2 F[][2] = {{1, 1}, {1, 0}}, Fn[2][2] = {{1, 0}, {0, 1}};

while (N) {
   if (N & 1)
      mul(F, Fn, Fn);    // Fn = Fn x F;

   mul(F, F, F);         // F = F^2;
   N = N >> 1;
}
```

2。另外一题很简单，但是蛮tricky的。How to test if a number "a" is power of 2

```
return (a-1) & (a) == 0;
```

[Coding Q1]: Given an array A, output another array B such that B[k]=product of all elements in A but A[k]. You are not allowed to use division.

其实这题interview之前在本版JHQ看过，可是当时看的题目太多，没有去想solution。所以刚开始听到这题还surprise了一下。我觉得这个不能用除法的限制太无聊了（建议改个problem来问这个algorithm），于是忍不住问why not division，顺便拖延一下时间想算法。面试官说除法慢...显然不是什么很convincing的理由，我说那乘法也慢啊。说完我已经想到怎样做了，于是顺利过关。

接着就来了比较衰的第二轮，题目是这样的：

[Coding Q2]: You are given a string e.g."face" and a set of mutation rules, e.g. a->@, e->3, e-E. Print all the possible strings that can be generated by the rules, e.g. f@c3, fac3, etc.

其实就是BFS再加上hash table判断是否重复print。马上就想到algorithm，面试官说好，你开始写吧。然后问题就来了，太久没写c++忘了hash table的函数定义。好像依稀记得hash table还有几个版本，想了一会没想起来，又不好意思问，汗！最后还是忍不住问了，他说你随便给个函数名和接口吧。最后磕磕碰碰总算把程序写完了，却给人留下了很不好的印象，感觉写程序很不熟！据说最后这个人给了我一个borderline，还算好，没把我fail掉。真惭愧啊，可怜我还是写c++起家的...

因为第二轮不太理想，本来应该两轮过后就onsite，结果hr来信说要第三轮phone，还很好人的说We do have three phone interviews at times. We are constantly evaluating our process so I apologize for the change. 第三个面试官又临时换人，最后居然是同系师弟，不过之前没见过面，不然可以套近乎了。。。言归正传，换人大概还是因为他们组想看看我match与否。他问的都是machine learning，风格和原来完全不同，还多多少少有点surprising的。

[Coding Q3]: Implement one step of decision tree which splits the node into two subtrees

1. How to call C++ code in C? How to call C code in C++?

2. In which three cases, initialization list has to or is preferred to be used for a constructor?

3. Can we design singleton by setting all the data member and method of a class to be static?

Some others:

4. Is overloading allowed in C? If not, how to differentiate them?

5. Default methods that are generated by a class in C++.

6. Difference of struct and class in C++

7. Given a class has first name, last name, SSN and etc.Need to query according to first name, first name + last name, what STL should be used? (If map/multimap, what should be the key?) How the query should be? How to query all the first name that initialed as "J"?

8. Meaning of static in C and C++

9. Meaning of inline in C++; where should it be used?

什么样的情况下用 virtual deconstructor？

2. virtual function是如何工作的？virtual table 是如何实现的？

3. virtual function具体调用哪个function是在编译的时候，还是在代码执行的时候决定的？

4. 类的copy constructor 和 assignment operator "=" 有什么区别？有什么主意事项？

给定一个二叉树的一个node，编程返回中序遍历的下一个node。如果最后一个，返回null, 怎么做？

一堆数，其中一些数出现了一次，一些数出现了两次，只有一个数出现了三次

找出那个出现了3次的数

hash方法很trivial就不说了。

如果用bitwise operator，怎么高效的做？除了XOR，是不是还得用点别的办法？

1. Mempool design with 30k limit.
   mempool是应该在一开始就allocate 30k 连续的内存,然后分配和管理？

   或者是每次call allocate(n)的时候再通过operator new[]来分配内存，update size member？如果是的话，free(ptr, n)怎么写呢？貌似operator delete[]不能带size参数啊？

   总之我就是对memory design这块很不熟悉。。。

   2. Implement put/get methods of a fixed size cache with LRU replacement algorithm.
   这个是不是用fixed size的max heap来实现？每个元素定义一个key，表示距离上次使用的时间，每使用一个元素，就相当于是把它的key更新为比当前最小值更小的数，然后做heapify()操作？
   每put一个元素，就assign新元素一个最小的key，然后用新元素替换掉堆顶点，然后做heapify?

   3. Write a function to implement a buffer for DataOutputStream.
   这个我完全没啥概念了。。。求指点一下~

   4a. How do you write malloc and free to detect memory reference violation?
   4b. flag a block of memory as used by putting some bit pattern at the beginning of the block. What bit pattern will you use?
   这两个问题也是摸不到头脑的。。。

   5. How to implement singleton without using static/global variable?

1. **say some http methods?**
   2. get/put difference?
   3. what does DTD for xml mean?
   4. common protocol used in layer 4?
   5. describe different ways to use css in html
2. **. difference between well-formed and valid xml?**

   前两天第二轮technical phone interview:
   1. why and how did u get into web development?
   2. what do u like about web development? not like about it?
   3. why do u want to work for google? 我扯到ajax的推广，他顺着问 ajax principle, security issue
   4. what language are you comfortable with? talk about it. why and how did people design it?
   5. explain 3 components of MVC
   6. what happens when a user types google.com in URL bar and press enter? (dns, http get, tcp connection establishment, etc)
   7. what may slow down the rendering of html

1. there are only 6 db connections in the pool, all 6 are being used, another request needs to connect to DB, it does not want to wait. How to do it? One solution is to make a new connection and add it to the pool. But the interviewer wants standby solution. Anyone knows the standby solution? Thanks

   2. For the db connection. min = 10 and max = 40.
   Will 10 connections be created at server start up time?

2.1 If we start with using 13 connections, when all the jobs are done, how many will be kept in the pool? 10 or 13?
2.2 If we start with using 45 connections, when all the jobs are done, how many will be kept in the pool? 40 or 45?

然后OOD的一道题，其实不难，但我感觉自己答得不好
have a furniture class, some child classes like table, chair, etc.
they want to extend the class hierarchy, as there are wood table, steel table, wood chair, steel chair, and so on.

我首先给出class + interface的design:
furniture (table, chair, ...)
table (woodtable extends table implements wood, steeltable extends table implements steel)
chair (similar as table)
然而interviewer立刻指出这样的话，如果要加fire, 或者和重量有关的functions时，会有code duplication (因为wood可燃，steel不可燃, assuming fire function is defined in wood and steel interfaces). 我最后说那把wood和steel也改成class (c++ multi-inheritance), 这样可以avoid code duplication. Interviewer又问那么如何实现woodsteeltable? 我说就inherit woodtable and steeltable. 自己对这个问题总体感觉不好。大家有什么好的design?
后面的问题比较简单：
given a deck of cards, how to shuffle it?
having a web application, front-end, middle layer and database. How to scale database to accommodate increasing traffic?

栽在一道编程题上：Find a longest increasing subsequence in an integer array。
问问题的人要求朋友拿出O(nlog(n))的算法

. one array filled with numbers from 1 to N, but one number is missing. what's the most efficient way to find the missing item? what about two or more numbers are missed?

2. find the repetative chars in a string and delete them

3. find the binary tree from its preorder and inorder traversal

given a character string, print the number of occurence of each charcater in order. ie. if the string is "ceabcw", then you should print something like:
a 1 b 1 c2 e 1 w 1.
she asked the possible data strucutre to approach. I gave array, hashtable, and BST. she asked me to use BST, and using no recursive. Also how to handle unicode.
1 下面的int * takeaddress()有没有问题， 啥问题？
2 写个效率高的takeaddress出来， 实现同样的功能

```
int  * paddress, address1, *r;
paddress = takeaddress(); /* defined below */
address1= paddress[0];

int * takeaddress()
{
   int  address[8];
/* The address are defined here */
  return  address;
}
```

一个字符串，要求返回重复次数最多且最长的子字符串（假设源字符串中最长重复次数最多的子字符串只有一个）。例如 “abcabcdfabcdf”要求返回“abcdf”. 因为“abcdf”重复次数最多且最长。
俺只想到两个土办法：
1)找到所有字符串组合（例如a, ab, abc, abca, b, bc, ...），都放入hash table

，找重复次数最多的且最长的。
2)用Dynamic Programming找LCS的办法，两个字符串都是源字符串，然后在那个2D array里面找最长match，并计算重复的次数，然后输出结果。

很多都是老题，不过我专门整理了一下：
1. string match:
string Text, Pattern;
find a substring of Text matches with Pattern.
解法纲要：Rabin-Karp, KMP, suffix tree

变种1b: multiple match:
string Text, PatternSet[n];
find a substring of Text matches with any one pattern in the set;
解法纲要: Rabin-Karp

2.LCSubstring:
string A,B;
find the longest common consecutive substring;
解法纲要：DP(A.len*B.len复杂度)，suffix tree(A.len+B.len复杂度)

3.Longest Palindrome
string A;
find the longest substring of A which is a palindrome;
解法纲要：类似2

4.Wild card match:
4a: Pattern contains '?'(s)
4b: Pattern contains '*'(s)
4c: Pattern contains both;

//以下是与dictionary有关的题目

5. dictionary + wild card search(一般都需要做适当预处理):
第一种search：search所有match结果
第二种：返回某个特定的结果，比如，所有match中最长的单词
5a: pattern = ??a????b* (指定某些位上的字母)
5b: pattern = abcde* (指定fixed/unfixed length的前缀)
5c: pattern = ?a*bcd*e?f* (?和*任意混合搜索)
解法：待探讨

6. dictionary + 包含字符集合:
Letter_Set = "aabbbcd";
第一种search: 所有至少包含2个a,3个b,1个c,1个d的单词
第二种search：所有至少包含这个字母集合的单词中最长的/最短的
解法：待探讨

7. convert a valid word to another valid word of the same length, by replacing one letter in one step, every intermediate word must also be valid;
解题思路：相同长度的单词构建一个图 + BFS

8. edit distance (misspell correction):
type a misppell word, give top10/all suggestions of correct words;
解题思路：首先定义计算edit distance的metrics，然后从每个valid单词计算出到它距离<=某给定值的所有misspell的单词(类似BFS的一层一层的算)

9. find a matrix with max area: each row and each column of the matrix must be a valid word;

10. 朴素搜索，在dictionary中搜索一个单词是否存在：
解题思路：hash; trie;
10b. shortest unique prefix: give a string, find its shortest prefix, which doesn't match with any prefix of any valid word in dictionary;

for example:
cat against {dog, be, cut} is ca
cat against {dog, be, cut, car} is cat
cat against {dog, be, cut, car, cat} is null
解题思路：trie/prefix tree;
11. solve a crossword puzzle;

1. N台机器，每台机器有N个数
   找median (2个数组找median的扩展版)

   2. 已知coin denominator set，例如,2cent, 3cent, 5cent...
   给定一个目标数，比如126cents
   最少需要多少个coin。

   这个题我以前问过一次，没人回。。。我觉得是很好的题，贪心，回溯，DP都可以试试。但是我一直没找到最满意的解。

   3. 一个整数数组，找3个数满足勾股定理。求比O(n^2)更好的解

c++ and data structure
** single linked list, find nth from the end
** Overwriting and Overloading
** Stack vs. Queue
** Array of integers, all integers appear even times except one, find the one appears odd times. (some following up questions for this one)

OOD
** Do you approve the following design?

```
Class Furniture{
      Some functions related to the property of furnitures;
};
```

4 classes derived from Furniture
Class wood_chair
Class steel_chair
Class wood_table
Class steel_table

What if you need to design a lot of other furnitures like desks.... with other materials like plastics

**An open question.Takes more time than any other questions. It is related to the project they are working on, you should not be asked.

1. given n strings with equal length, say x. find the substring shared by all of them. For example, abcx, abdx, abea, then ab is shared by all of them.
   2. the gmail page loads very slow. any suggestion for improvement?
   3. we want to check the number of querys obtained from the world in the last minute and the last hour, what data structure should you use for that? If there are billions of records, i.e, too many records for the main memory, what suggestions do you have?

你有一种语言的dictionary,你有一大串string,没有delimit,你如何interpret成字典中的字呢？

Given a binary tree

```
struct node{
struct node* leftChild;
struct node* rightChild;
struct node* nextRight;
}
```

The nextRight points to the right node to the current node in the same level. Ask you populate the nextRight pointers in each node

1. Java里如何比较两个objects是否相等
   2。怎样找出一个list是否包含循环
   3。inheritance和composition：什么时候需要用到哪种？
   4。一个int array
   如何找出subarray，使得元素之和最大

1. 给定一个首尾相连的排过序的单链表，首节点最大尾节点最小，给出链表中任意一个节点，要求返回链表中间节点；

   2. 一摞未排序的扑克中间有重复，用最有效的方法找出并删除重复者

1. how to find 1 missing number from 0 to N in an array of N numbers.
   2) brainteaser, 5 jar problems.
   3) how to calculate sqrt(N) without using sqrt function. Binary search tree problem.
   4) some behavioral problem. Like, How do you know about BB? Why you wanna work in BB? Why you wanna work in industry?

   3. onsite interview, Jan 2010
   1st meet 2 people in R&D
   1) train, tunnel, people escaping problem
   2) 6 digits number, each changes from 0 to 9. Find the odds that sum of first
   three is the same as the sum of last three. A: 2 do loop.
   3) Find 1 missing number from 0 to N. But notice that it is possible the sum would overflow. Think about a way to avoid the overflow.
   4) Tricky problem. I do not think anyone else would know the answer except the one who gives the problem. Nothing to do with math, statistics.
   5) Same 5 jars problem. That is their favorite.

.behavior question, Why you want to join BB?
2.一个windows系统，一个unix系统，unix系统里有100个数据库，总共1TB，如何在1小时内从unix系统转移到windows系统中
3.找出一个字符串中最早出现的非重复字母
4.两个鸡蛋测试那层楼丢下来会碎

. 用两个stacks来实现一个queue，题不是很难，但是要求逐行念代码，精确到冒号分号，尖括号怎么说
不知道。。。耗了好久。
2. 一些关于multi-threading，critical section，等等。
3. SQL的一些问题，我不怎么会，就skip了。
4. OOD问题，如何设计parking garage，大家有什么好的想法吗？
//
1.calculate the Depth of Binary Tree
2.Graph Connectivity(Adjacency List)
3.An array of integers, only one integer appears odd times, all others appear even times, find it
4.Process Vs. Thread and Memory allocation(stack and heap)
5.How to check singly linked list is a circular linked list.
6.Override Vs. Overloading
7.check a number whether is Power of 2
8.Given an array of integer and a target number. please find out two number that add up to the target.
9.Factorial and optimization
10.Design a file system
11.Check whether a binary tree is BST(if the binary tree is very large,

you can not simply in-order print all the nodes out.)
12.Design a Crossroad signal system
13.implement atoi()
14.Pattern Matching, if '.' is used as a wildcard, which means '.'can represent any character.
15.Design a game(algorithm): which transform a word to a target word. for example: from head to tail, each step, you just can replace one character, and the word must be valid.
16.In a clock, calculate the angle between hour and minute handle
17.Given a singly linked-list, and a pointer to node, how to delete the node.
18.File external sorting

C: pointer, call by value/pointer, return the pointer of a local variable, string manipulations, source code of some important C string subroutines (strcpy, strtok, etc), itoa, atoi, static variable and fuction, name mangling,
memory allocation
http://www.eskimo.com/~scs/C-faq/faq.html

C++: namespace, abstract class, polymorphism, dynamic binding, virtual function, virtual destructor, constructur, copy constructor, assigment operator, throw vs exit, C vs C++, Java vs C++, static attributes and methods, multiple inheritance vs interface, struct vs class, templates, template specialization. 如果你再会bit operations, union就更牛了。
http://www.icce.rug.nl/documents/cplusplus/
http://www.parashift.com/c++-faq-lite/

OS 基础：常用shell commands, like top, ps, sort, jobs, diff, find。Synchronization， deadlock, race condictions, mutual excusive, semiphore, signals, multiple processing vs multiple threading。Communication between processes or computers, IPC, pipe。

Data stucutures and algorithms: array, largest N integers, find duplicates, missing value. linked list (必考！). sorting algorithms and complexity analysis, space vs speed. hash table, array vs linked list, linked list traversal, reversion, insertion, deletion and sorting, loop in LL. String reversion, remove duplicates, reverse words. Tree traversal, traversal without
recursion, breadth-first traversal, compare two tree, find common ancestor.

Create a command-line program (GUI not necessary) in C++ to do the following:
•Takes in 3 arguments from the command line:
o Input file name: The input file will contain several rows of comma-separated name/age pairs; one pair per line (see below)
o Operation type: one of 3 values specifying what type of operation to perform on the entries in the given input file:
“n” for sort by name
“a” for sort by age
“f” for filter out entries whose age is less than 21 (no sorting required)
o Output format: one of 2 values specifying how to format the output of

the operation.
    “h” for HTML table.
    “c” for comma-separated values (i.e. the same format of the input file)

•Has one output:
- o The output should be the sorted or filtered list of name/age pairs, in the output format requested; dumped either to stdout or a file.

Example input file contents:

```
Jim, 50
Abe, 20
Mike, 33
John, 21
Abe, 18
```

First phone interview:

1. what's the difference between overloading and overwriting
2. what is polymophism?
3. Design pattern--singleton
4. design elevator system
5. An array which has some elements (each element appear even number of times except one, find the one that occurs odd times)
6. coding -- convert $134.35 to one hundred thirty-four and thirty-five

Second phone interview

1. talk about the data structures you know
2. talk about sorting algorithms you know
3. Design question: furniture, woodchair, steelchair, woodtable ...
4. Algorithm:
   1) Exchange two variables without using a temporary variable
   2) Find elements in an array that sum up to a given number

不敢保证全部涵盖，大部分的都在。
我自己找了一遍，大家一起用着都方便。
不过只是含有题目的帖子 我才包含进来了，只分享经验没贴题目的 我都没有包含进来。

大家复习着方便。

1. 一个 sorted interger Array[1...N], 已知范围 1...N+1. 已知一个数字 missing。
   找该数字。
   把原题改为 unsorted，找 missing 数字。 performance。

2. 复制 linked list。 已知每个节点有两个 pointer，一个指向后一个节点，另一个指向其他任意一节点。 O(n)时间内，无附加内存，复制该 linked list。（存储不连续）

3. 一个 party N 个人，如果一个人不认识任何其他人，又被任何其他人认识，此人为 celebrity。用 O(n)时间找到此 celebrity。

4. 给中序后续，构建树。

其他的每轮都问了简历。

感觉答的都不错，没什么难度。不知道为啥就被拒了。总之感觉很奇怪，不过也无所谓了。
希望对大家有帮助。

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31342084.html

>

贡献一道 cs 面试题，虽然我的面试机会极少。：D

设计一个函数，返回一组数字的组合，combination，
不同的是，你每调用它一次，它就返回下一个组合，而不是一次全返回。
注：你不能一次算完，把他们存起来，而必须临时算！

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31347263.html

>

IT company, 全都是 brain teaser, 有点老。

1. 50 个黑球 50 个百球，2 个罐，要求你放这 100 个球在这 2 个罐，使得别人随机从 2 个罐中任意拿一个球是黑球的几率达到最大。

2. heard on the street 上的男人出轨题，简单逻辑推理。

3. 这个没答上来，后来给了提示做出来了，但是回头想想还是不对。上来请教一下
。

2 个人商量好策略，然后一个从 52 张牌里面随机抽 5 张，看牌，考虑。。。然后排在
桌上，摊开前 4 张，第 5 张面朝下，由第二个人判断第 5 张牌。 问这个策略。

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31347264.html

>

这周一面的，一共 5 个人，碰到的题目不太难：

1. 写 atoi 函数，
  测 nodepad

2. 古老的三角形问题：输入 3 边，看是什么三角形。
  一个 mobile device 可以从服务器上传和下载图像，怎么测试这个系统？

3. lunch meeting 之后回办公室打开电脑，说他们现在开发的某产品有问题，每次要 loading 很久,差不多 10 秒的样子。问怎么测试并找出这个 bug？ 这个把我难住了，胡乱讲了一通，然后说太困难了；于是他换了个题目，画了一个 plotter 软件的界面，问怎么测。

  coding 的题目是 Path Walk，给一条路径，写一个函数来走通它。其实这个题目我没搞明白什么意思，先沟通了很久，最后开始写（还是不太明白。汗...），写完了觉得不正确，正想再改改，被打住了，说给个 test case 一起来看看程序怎么执行。每句代码跑了一通，却发现 code 写正确了:-)

4. 一开始是个 IQ 题，把一堆数字填到格子里，满足一些条件，比如 1 和 2 不能相邻。
  测一个记事薄软件。有 scheduler 和 notifier 两部分，可以从 scheduler 输入时间和内容，然后 notifier 到预定时间会给出提醒。
  coding 题目很容易，找到单链表倒数第 N 个节点。

5. 最后是 hiring manager，问了一些 behavior 问题，然后打开一个网站，问怎么测试。

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31348374.html

>

在 onsite 面试中实际遇到的。

1.template 中用 typename 和用 class 有什么区别？

2.unix 下执行 shell 脚本和执行可执行文件有什么区别？哪个更快，为什么？脚本语言程序（如 javascript)和可执行文件程序有什么区别？shell 和这两者却别呢？

3.如何对 const data member 做 assignment?

```
class A{
const int a;
public:
   A():a(0){};
   A(int m_a):a(m_a){};
};

int main(){
  A a(1);
  A b;
  b = a; //how to implement assignment for this?
}
```

4.如果把 base class 对象赋给 derived class 对象,会怎么样？compiler 报错还是执行错误？

```
class A{
public:
  int a;
};

class B:public A{
public:
  int b;
};

int main(){
   A a;
   B b;
   b = a; //what happend?
   cout << b.b << endl;

   B* b2;
   b2 = &a;  //how about this?
   cout << b->b << endl;
}
```

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31348607.html

>

两波两小时，第一波老白＋老印，小兵

1. 介绍 Recent Project
2. 两个 C 的程序问题
先是 char*指针问题

```
char *dosth()
{
char s[256];
char * p = r;
p = "some new string":
}
```

然后问了一堆变量的值，比如 s, *s, *(s+2), &p, etc.

另外一个 switch 程序找错，没有加 break 之类，还有就是 return local variable 地址的问题

3. 手写 fab(n)函数，不是算，而是输出，递归或者循环都可，不过递归不高效大家应该知道

4. 逻辑问题：八个水罐称重

5. 一堆关于 OO 概念的问题，多态，继承，封装，接口和抽象类的区别，复写和重载（包括 C++具体怎么实现的）

6. 反馈问题

第二波一个项目经理
一来就是比较高难度的，给你一个字节数组（注意取值范围），数组长度可能非常长，如何找到第一个只出现了一次的数字。开始没什么思路，和他讨论了一会，边问还边问复杂度和数据结构的问题，后来发现应该进行数出现次数，这样复杂度就是２ｎ，结果出来了要求手写出代码。

然后就是一个智力问题，三个囚犯黑帽白帽，之前没见过，所以用了不少时间才想出来，大家可以搜搜，有现成的。 最后反问问题后结束。

虽然每个问题都答出来了，不过最后还是功亏一篑。不是很清楚什么原因，哪位同学能够指点一下？现在继续回归闺中状态中。不过感觉彭博问题的重复概率很大，希望对后来的同学能够有所帮助：）

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31350186.html

>

1, C vs C++
2, struct in C v.s. in C++ v.s. class in C++
3, virtual function, pure virtual function, abstract class
  what is the advantages of using virtual function
4, new v.s. malloc()
5, memory for a process (code, static data, stack, heap)
6, how to know the stack is growing in the direction of address increasing
or decreasing
7, virtual memory

Pasted from <http://www.mitbbs.com/article/JobHunting/31354737_3.html

>

nsite 只见了 2 个人，估计 over 了，总结一下教训。
题目不难，主要一共 3 道题，都算比较基础的题。
1.reverse words in a sentence，使用如下函数。
char* reverseWord(const char* str)

2.an interger array containing millions of elements with min 0 and max 1000,
how to sort it?

3.covert interger number to date string, for example, 20090130 -> "01/30/
2009"

说说教训：
第一道被输入 const 给搞死了。先是没有注意 const,直接按照常规非 const 做，没有写完就被叫停了；然后是被平时强调的 malloc 后必须及时 delete 规则搞死，坚持认为在函数里 malloc 一块内存然后在函数外 delete 是不好的习惯；最后当面试者提出如果定义一块内存
，如 char tmp[2048]，然后使用会怎么样？自己提到可以在函数外 strcpy 函数返回结果
，却忘了
arr 大小实际是无法指定的，所以这种方法是不可接受的。总之，很多的 trick 在里
面没有注意到。

第二道使用 couting sort 应该就可以。面试者要求描述算法，不需写代码。

第三道自己的做法是先取得 30,然后 01，然后 2009 然后组合成一个 string。问题是这样的话，月和日的 01 前的 0 可能会丢失，使得最后结果可能不对。后来想正确的做法应该是先把整形转成 string,然后使用 substr 并组合。另外，面试者问有 stl 有什么可以替换 itoa，自己答不出来。后来查了下，应该是可以使用 stringstream 来实现，如下：

```
stringstream ss;
ss << intVal;
```

ss.str()就是我们要的结果。

Pasted from <http://www.mitbbs.com/article/JobHunting/31356298_3.html

>

现在版上的文章好像已经以 H1 申请为主了，不知道多少人还对面经感兴趣。不过想想还是写一下好，要不自己过几天可能就忘光了，呵呵。

申请工作：Software Developer，Mobile Application 组

问题问的很多很杂，基本上不难。不过后来想想，也没有全都答对。大概包括：

* 为什么对 Amazon 感兴趣。
* 自己最近的 Project。
* 说出自己会的编程语言并打分（1－5）。
* 有没有开发 Mobile application 的经验。
* 几个常见 Data structure 的 Lookup 操作的时间复杂度。
* HTTP post 和 get 的区别。
* Design Pattern: Singleton, Factory, Lazy initialization。
* Multi-threaded programming, deadlock 之类。
* 对 Unix 环境是否熟悉，几个常见命令，ls, ps 之类。
* Reflection 的概念，Java reflection，C++里面是不是有 reflection。
* 如何实现 Garbage Collection。Reference counting 的缺点(cycle)，如何解决，JVM 有没有解决。
* C++里面 virtual destructor 的用途，于一般 virtual function 的区别。
* 写一个函数实现两个整数相除，不用"/"和"%"，返回商和余数。写完读给他听。
* 算法设计：一个 Galaxy，每个星星用一个三围座标表示，找出离地球最近的 1000 个。（后来发现这也是老题了）

Pasted from <http://www.mitbbs.com/article/JobHunting/31368921_3.html

>

have read many posts here although not really helpful this time, here is
something about bloomberg's phone interview question

1. How to implement garbage collector (what data structure)
2. How to implement c++ smart pointer
3. Pro and Con of multi process and multi-thread
4. How many stack/heap does a multi-thread program with 10 threads have?

there are some other questions I don't remember.

☆──────────────────────────────────────☆
yellpine (fresh CS master looking for referral) 于 (Wed Mar 4 16:42:52
2009) 提到:

4. How many stack/heap does a multi-thread program with 10 threads have?

who can help me with this?

10 stacks? 1 heap?

Pasted from <http://www.mitbbs.com/article/JobHunting/31373641_3.html

>

估计没有多少人感兴趣,

不过还是说一说.

面的是 ELASTIC COMPUTING CLOUD 组的 SE,我是做网络的,对他们的这个 cloud service 有些兴趣,以为会问算法和系统的问题,结果问了一个 OOD 的问题,

说一个大楼,10 层,4 个电梯,怎么设计类来实现这样一个系统?

题目 career cup 上有,不过没想到他会问这个,ECC 又不是做应用的.刚好是我的弱项,一直做 research,对算法和语言还算了解,对应用系统和设计那是一片空白.面的是一塌糊涂. 有要面 amazon 的参考一下.

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31376671.html

>

S Master
Windows Live Experience 组：
我一直催 HR，今天给我打电话说拒了。 汗了--自我感觉还挺好的。

拒了 也把经验给大家分享一下，在版上受益这么多，回报一下大家。
一共见了 5 个人， 两个 SDET， 两个 SDET lead， 最后一个 director。题目都很简单

1.美国人。上来随便聊聊，然后出了个 coding 题目 一个数组，找出第一个重复的数
 我给了三种方法，最后用 hash 写的，然后问 test case 之类的

2.印度人
 上来问我会什么 C#还是 C++，我说 C#会的多一些。然后他上来问了四五个简单的 语法问题。正好我还都会，心理还窃喜着呢。coding 也很简单，给一个 01 字符串，转化成整数。写完后 test case。 第二个题目是两个函数互相调用，无限循环了，然我找出毛病，问怎么解决。
 然后午饭跟这个印度人吃，随便聊聊。 就过了

3.欧洲人，不知道哪国。
 女的， 人很好，跟她聊的最开心。coding 题目是个没见过的，double bytes string 实现 delete 键功能。这个比较难解释，她开始也跟我解释了很长时间。 就是删除字符的时候如何确定是删一个字节还是删两个字节的问题。我给出算法，然后她有提示有哪些特殊情况要考虑，也做出来了。然后她就问我给一个一般的 application 如何测试，又随便说了一通，结束了

4.美国人 senior test lead
coding 很简单，给一个句子，把里边所有的单词自身 reverse
然后给我看他们的产品，问我怎么测试。聊的也挺好

5.欧洲人 director

面到这个人的时候，我都快累趴下了，都不想面了，实在是累。心理还想着，offer 拿不拿得着无所谓，别把老子给累死了。（看来真得努力锻炼身体，不然面试都挺不住）

题目也很简单，找 1--100 的素数。我就给了最简单的方法，然后我说要 check 一些边界情况，他说不用了。然后让我做到他的椅子上，打开 excel，问我怎么测设置字体这个 feature。说完了问我有什么问题没有

面完后觉得还不错，因为去之前心态还不错，现在就业情况不好
也怕，没打算毕业。
后来回到学校我就一直催 HR 赶紧给我结果，今天就给催出来了 呵呵
我说打算 postpone 到年底毕业，他说到时候可以直接再联系他（有个屁用啊 呵呵）

在学校继续呆着吧，呵呵，把身体锻炼好，面试也是一个体力活。
写出来 希望对大家能有点帮助。

Pasted from <http://www.mitbbs.com/article/JobHunting/31383513_3.html

>

iantianz (tiantianz) 于 (Thu Mar 12 15:23:30 2009) 提到:

刚刚电面了一家中型软件公司的 summer intern，问了一个算法题：

给你一本 dictionary，任意给你七个 letters，让你找出包含这七个字母的、最长的单词。

条件：可以 pre-processing，这样每次给你不同的 letters 时，可以 very effcient

我当时想了好久也没给出完整答案。。。
naive 的解法当然就是每次 scan dictionary，每次 O（n）。。。
pre-peocessing 那就是建 index，但 index 怎么建？怎么操作？

Pasted from <http://www.mitbbs.com/article/JobHunting/31387661_3.html

>

MichaelChen (Michael) 于 (Thu Mar 12 17:05:44 2009) 提到:

上学期 ON CAMPUS 的 INTERVIEW,PENDING 了几个月给了 ON SITE
两周前去的，面的 FULL TIME SDE。
见了 5 个人
1.REVERSE LINKLIST.
2.给了 N 个数，值域[1,N-1]，如何找出第一个重复的数
3.算 POLYNOMIAL，比如 5x^4+6x^3-7x^2-8=?
4.给一个 URL,如何把空格这种字符转换成%20 这种
5.给一个 LINKLIST,VALUE 的指针指向其他 NODE，复制他
今天 RECRUITER 发邮件通知给 OFFER 了，漫长的两周。。。希望收回 OFFER 这种事不要发生。。
准备基本就是看 MITBBS 精华区，programming interview exposed 和 careercup，希望大家都好运，ms 应该还是在招人。

Pasted from <http://www.mitbbs.com/article/JobHunting/31387663_3.html

>

攒 rp，最近遇到的面试题。。。

C++：effective c++上的东西若干；exception 相关；继承和子父类指针若干. 十五分钟左右。

算法/编程：1. 大文件随机 sample，one pass. 2. sodoku solver. 3. logn 解 x^y,
4. DP 题 5. 1Billion query 里选出时间最近 5 分钟内最 frequent 的 1000 个，one pass（我以前在 amazon 见到过这题）。6.两个排序数组找共同中值。递归和非递归解法。7. 斐波那契数列。100 层楼梯下楼，可以一步也可以两步，多少种下法？递归和非递归。8 贝叶斯后验概率。9。多少人在一起，生日可能出现重复概率大于 0.5？（算法导论原题，我只记得个答案，直接说了。。。）10. 一个数组，找最大值比较次数？同时找最大值和最小值比较次数？找最大值和次最大值比较次数？（他问我是否知道这题，我说是作业题。后来和师兄聊说是这他常拿来用的面试题。）

系统设计和经验：1 设计一个库，提供 timer 的功能。deltalist/hash，或类似 linux kernal 的 timer 设计。效率要比较高。2. 一个类似 chord 的 DHT 设计。3. 你有一个奇怪的程序，有时有 bug，有时没有，说出尽可能多的可能原因。4. printf 来 debug 有何不妥。5. process 和 thread。process 之间的 IPC 有那些种？process 间是否也可以 share memory.何时选 thread 或 process。

Pasted from <http://www.mitbbs.com/article/JobHunting/31393101_3.html

>

职位是 junior financial engineer, 公司是一 hedge fund,其实面完就感觉不太好，
一共见了 6 个人，有两个人问得技术问题答得不太好，也怪自己事先准备面试下的功夫不太到家，准备得重点没有把握好。以下是一些能想起来的问题：

1。C++ 中的 virtual destructor 是啥？ 为啥要用？

2。quick sort， merge sort 的复杂度。

3。 Structure 和 class 的区别是什么？ （我晕，这个我居然给答反了）

4。关于 C++ 处理异常的方法 。 （基本上一头雾水）

5。Monte Carlo method in american style option pricing。 （我说的用 least square regression method,blah......)

6. Int_0^T W(t) dW(t) （ 一看见这个，贼激动阿，熟悉的 ito' s formula)

7. Stonivich intergral 是啥？ 为什么用 Ito's 不用 stonivich? (不知道拼得对不对）

8. 一个国家所有的人如果生了一个男孩以后就停止生育，生了女孩以后就继续生，直到生出男孩才停止生育，问多年以后男孩多还是女孩多？ （要联系上 stopping time 的概念）。

9。什么是 AR model? 啥时候用 AR model?

10. American option 的 up bound? (我说是 stock price,被直接鄙视了，说更精确的
，只好答没有研究过，当时一头雾水）。

以上的是我能记得的一些问题，还有一些关于数据结构的问题我就彻底歇菜了，从来没有这方面的背景，也就不记得他的问题了。

还有就是，关于自己的简历上面的 Project 工作经历，一定要熟练再熟练，有些人问得那叫一个细啊，而且基本上我所有的 Project 都被人问到了。这次面试的前 4 个人主要问计算机和金融方面的技术问题，第 5 个 HR，问些 personality 的问题，最后是 hiring manager,因为之前电话面试过我，就没有问问题，简单聊聊。整个面试花了 5 个小时，雷死了，脑子到后面都已经不转了。虽然结果让人遗憾，不过就当是学习了，贴点信息和大家共享下，希望自己能早日找到工作，也希望还在努力找工作的 XDJM 们再加把劲，大家一起加油。

Pasted from <http://www.mitbbs.com/article/JobHunting/31406731_3.html

>

CS 方向，希望对大家准备面试有帮助

1. 用 stack class 来实现 queue，具体用几个 stack 不限。完了以后问怎么实现 thread safety，然后是怎么测试。

2. 实现 strstr(str1, str2)，如果 str2 是 str1 的子串，返回 true，否则返回 false。实现完了以后问如何测试。

3. 给定一个 integer array with both positive and negative numbers，return a contiguous subarray with the largest sum. 我本来想用 dynamic programming 实现
，但面试官希望按照他的一个更 heuristic 的思路来解，最后勉强搞定。

4. 给定一个排好序的 linked list，删除其中所有的重复元素。比如给定 1->2->3->3->

4->4->5，返回 1->2->5。给定 1->1->1->2->3，返回 2->3。看起来简单，一边写一边发
现许多细节需要小心应对，好在最后搞定。

5. 给你三个烤箱，每个烤箱可以同时烤两片面包，需要的时间分别是 3 分钟，4 分钟和 3 分钟。但第三个烤箱有一个 slot 出了点问题，每次只能烤面包的一面。所以这个烤箱三分钟后只能算烤好一片半面包，你需要把那半片翻个面，在同一个 slot 里再烤一次才算一片完整的。现在给你这三个烤箱，问烤好 21 片面包最少需要多少时间？如果是 2100 片呢？如果是任意给定的 N 片，要求 O(1)时间内给出最少需要的时间。

6. 给你三根棍子，每根都需要一个小时才烧完，但每根燃烧的速度都不一样，也不均匀。问只有这三根棍子和火柴，如何精确的得到 1 小时 45 分钟的计时？

7. 在一个 party 上，每个人可能认识别人，也可能不认识。现在其中有一个人是名人，定义就是所有的人都认识他，但他不认识其余的任何人。现在要求你去找出这个名人来。但你只可以通过一个方法，就是问 A 是不是认识 B，回答是表示 A 认识 B，不是表示 A 不认识 B。你可以任意去问这样的问题，问最少需要多少次能找出这个名人？思路有了之后要求写代码实现，可以调用 knows(A, B)，代表上面的那个问题。实现完了以后问如何测试

8. 测试 copy 这个命令。然后自己问了一些 clarifying questions，搞清了实际上是
copy src dest。src 可以是文件，也可以是目录。dest 可以存在，也可以不存在。

Pasted from <http://www.mitbbs.com/article/JobHunting/31410833_3.html

>

OO 设计题，怎么做一个十字路口的 traffic light.
2. 怎么不用 recursion 做二叉树 in order 遍历。

Pasted from <http://www.mitbbs.com/article/JobHunting/31421129_3.html

>

1. Write a function that returns a node in a tree given two parameters: pointer to the root node and the in order traversal number of the node we want to return. The only information stored in the tree is the number of children for each node.

2. Input a message and a text, find if the message can be composed by the text.
If the text is in a magazine (two pages/a paper), how to design an algorithm ?

Pasted from <http://www.mitbbs.com/article/JobHunting/31422009_3.html

>

1

When casting an object of a polymorphic class from a base calss type, which one of the following casts
performs the task only if the cast is valid?

a. static_cast
b. (void*)
c. dynamic_cast
d. const_cast
e. reinterpret_cast

```
2.  class A
{

public:  void f();
protected: A() {}
      A(const A&) {}
};
```

why are the default and copy constructors declared as protected?

a. to ensure that instance of A can not be created via new by a more derived class
b. to ensure that instance of A can only be created by subclasses of A
c. to ensure that isntance of A can not be copied
d. to ensure that A cannot be used as a base class.
e. to ensure that A cannot be instantiated on the stack

```
3. template<class T1; class T2; class T3>
int Product(T1 a, T2 b, T3 c)
{
   return a*b*c;
}
```

what is wrong with the sample code above?

a. templates must be class definitions
b. the template parameters should be separated by commas.
c. the template definition is missing a pair of braces.
d. template parameters must be pointer types.
e. the * operator has not been defined for T1, T2, and T3.

```
4. class FOO
{
   char * buf;

public:

   Foo (const char *b = "default")
   {
      if (b)
       { buf = new char[std::strlen(b) + 1];
         std::strcpy(buf, b);
       }

      else

         buf=0;
       }

   ~Foo() { delete[] buf; }
};
```

```
Foo func (Foo f)
{
   return f;
}
```

when the function fun is called, the program may crash or exhibit unexpected behavior, what is the reason ofr this problem?

a. the destructor may attempt to delete the string literal "default"
b. the destructor needs to check that the value of buf is not 0.
c. the class does not allocate a long enough buffer.
d. the function needs to return Foo& instead of Foo.
e the class needs to specify a copy constructor and assignment operator.

Pasted from <http://www.mitbbs.com/article/JobHunting/31426509_3.html

>

1．请书写一个程序，将整型变量 x 中数字左右翻转后存到另外一个整型变量 y 中，例如 x = 12345 时，y 为 54321，x = -123 时，y 为-321。其中 x 的个位不为 0。

```
    void reverse (int x, int* y);
```

(1) 请实现该函数，以上函数原型是用 C 语言写的，你可以用你熟悉的语言；
(2) 请写出一段代码验证该函数在各种情况下的正确性。

2．对集合{1, 2, 3, …, n}中的数进行全排列，可以得到 n!个不同的排列方式。现在我们用字母序把它们列出来，并一一标上序号，如当 n=3 时：

    0．123
    1．132
    2．213
    3．231

4．312

5．321

现在，请书写一个函数 void print (int n, int k)，（函数原型是用 C 语言写的，你可以用你熟悉的语言）在已知 n 和序号 k 的情况下，输出对应的排列，并简要阐述思路。

3．一维数轴上有 n 条线段，它们的端点都是已知的。请设计一个算法，计算出这些线段的并集在数轴上所覆盖的长度，并分析时间复杂度。例如，线段 A 的坐标为[4, 8]，线段 B 的坐标为[1, 5.1]， 那么它们共同覆盖的长度为 7。 请尽量找出最优化的算法， 解释算法即可，不必写代码。

Pasted from <http://www.mitbbs.com/article/JobHunting/31428195_3.html

>

Given a sorted integer array and a number, find all the pairs that sum up to the number.
这个很简单，但现在多了一个条件
What if the array is sorted by absolute value, for example {1, -2, 4, -9}, find the answer in O(N).
这样有什么好的思路么？

Pasted from <http://www.mitbbs.com/article/JobHunting/31430593_3.html

>

How do you find sequences of consecutive integers in a list that add to a particular number.
Array 里面正负数都有.
这个能在 O(n)时间内解决吗？

Pasted from <http://www.mitbbs.com/article/JobHunting/31431861_3.html

>

A m*n matrix of integer, all rows and columns are sorted in ascending order.
 Find the most efficient way to print out all numbers in ascending order.

Pasted from <http://www.mitbbs.com/article/JobHunting/31434325_3.html

>

一次面世 Google，问到 hash table 是怎么实现的。我说了一个取尾数(round)的方法
，他说这个方法很 navie，工业界一般用其他的方法，比方说 STL 的 map。

我想了半天没有想出来，到这里问问。hash table 具体怎么实现的啊？

Pasted from <http://www.mitbbs.com/article/JobHunting/31434401_3.html

>

49 辆赛车. Assume for each one, it travels the track in the same amount of t
ime every time. Also assume no two finish the track in the same amount of ti
me. Suppose you have 7 tracks, but no timer. Design races to find the 25-th
fastest with minimal number of races.

Pasted from <http://www.mitbbs.com/article/JobHunting/31434523_3.html

>

ime span: 38:39
interviewers: 2 -- 1 indian, 1 american

How do you know the bloomberg?
What position do you expect?
What language do you want to answer with? (I choose C.)
What kind of questions do you meet for the online assessment?

what is static in C? how is it implemented by the compiler?
write the definition of a function that returns both the max and min.
why do you use the condition variable?
how to implement a lock?
Under what condition must you use linked list instead of array?
what data structure can you use to store elements dynamically and access
them efficiently?
The complexity of finding any element in a linked list in the worst case.
multi-thread library programming: did you write your multi-thread library
with p-thread? is there any problem you have with you library?
did you do your projects on linux? If you want to find a string in a file,
what command should you use?
do you know vector in C++?
a question about real-time programming (I forgot)
what is buffer overflow?

一些问题是针对我的简历里面提到的内容，所以，简历里面的内容要尽量的吃透。

Pasted from <http://www.mitbbs.com/article/JobHunting/31434685_3.html

>

Given two classes:

```
class B
{
public:
B(args_1);
B(args_2);
// and many constructors with different arg lists
};

class D : public B
{
public:
D(args_1) : B(args_1) {}
D(args_2) : B(args_2) {}
// and many constructors with different signatures similarly implemented
// some additional stuff specific to D
};
```

Assume that the arg list for B's constructors are quite long and may be revised pretty often in the future, in which case D's constructors have to be recoded correspondingly. Duplicating the update by copy-and-paste will certainly work here. Can you propose a better way so that the update can be done in one place without copy-and-paste duplication?

Pasted from <http://www.mitbbs.com/article/JobHunting/31434891_3.html

>

Given a large string (haystack), find a substring (needle) on it.
感觉这道题不就是scan一遍吗？
有什么time and space complexity上面的trick吗？

Pasted from <http://www.mitbbs.com/article/JobHunting/31435419_3.html

>

准备了很久，看了很久算法的书。。 结果被问了一个怎么 optimize memcpy()..傻眼了。。碰到了女老印，倒霉～～～～

Pasted from <http://www.mitbbs.com/article/JobHunting/31435587_3.html

>

规问题后，问了个 code 的问题：

给一个 substr，如何判断它在不在给定的 str 里面。

substr 有两个新的符号可能在里面：
（1）* : 0-n 个任意字符
（2）? ： 1 个任意字符

太紧张了，所以面试者简化了题目，说去掉“？”，然后让 code 和测试：基本框架出来了，但是好多特殊情况没有处理到，比如 substr 以“？”起头。

后来又问如果加入“*”有没有思路，刚说了两句就 out of time 了。

唉，失败了，不知道有没有下文。

Pasted from <http://www.mitbbs.com/article/JobHunting/31436721_3.html

>

给定 X[1..n] and Y[1..m] 两个 arrays，已经 sort 好了.
如何找到 X <Union> Y 的 median？我说用 merge sort，要 O(m+n/2)时间。面试官明显不满意。
这个已经 lineal 了？难道还有更快的？

Pasted from <http://www.mitbbs.com/article/JobHunting/31437417_3.html

>

given a 32 bit number N and 2 numbers(A & B) that determine 2 different bit
pos
itions of N how do you make all the bits between A and B equal to another
given
integer k.
given (A,B is in the range [0 to 31] and
k<=2^(B-A+1) (so that k fits between B-A+1 bits). Give an O(1) solution for
th
is
e.g if N=9 (1001) ,A=0 ,B=2,K=5(101 then the result should be 1101 (1.e 13)

这个题是什么意思啊？

Pasted from <http://www.mitbbs.com/article/JobHunting/31437907_3.html

>

在做 careercup 上面的题目, 有两个问题没有看懂, 希望有人指点下

1 一个 BST, 给定一个值, 打印出所有的 path,使 path 上所有节点的值等于给定值;
2 一个 tree, 如何高效的找出最长的 path?

☆────────────────────────────────────☆

mitbbs59 (59) 于 (Fri Jul 3 15:35:37 2009, 美东) 提到:

这都是 amazon 的题目吧
1.sum of all nodes in a path = givenValue
2.http://www.careercup.com/question?id=87897

Pasted from <http://www.mitbbs.com/article/JobHunting/31441709_3.html

>

是现场写 code 的面试。
第一道是写一个函数，两个参数（String prefix, String s）, 返回 true 如果 s 有
prefix
第二道是写一个函数，两个参数（int[] a, int sum), 找出数组里加起来是 sum 的几个数
我第一题算是答出来了，第二题没做完，没有好的思路。。。
大人指教一下

Pasted from <http://www.mitbbs.com/article/JobHunting/31446979_3.html

>

Went to Adobe to interview a Senior SW Engineer position,

总的 interview 的不错， 但被下面问题问倒了，让回去想想，

Q1:
"We need to compare thousands text files with each other, they are not big, less than 100K each. They are in a directories tree, with a few levels of subdirectories, how to speed up the comparing process ?"

My answers: We can read them all of these files into memory once so that we can reduce the number of diso I/O.

[Feedback: That is a good appoach].

Q2: How to read these files into memory (on MS Windows platform) ? how do you maintain directory structure in memory ?

My answer: I talked some garbage ....

Q3: If someone already wrote the code in slow way, read each file from disk, do some thing, close the file, read another one, etc. Can you make a " portable layer API" libary so that with minimal effort, old code can still work but much faster ? (of course, we need to recompile the code).

Please help with Q2 and Q3, thanks a lot.

--

Pasted from <http://www.mitbbs.com/article/JobHunting/31448285_3.html

>

今天把 M 的 onsite 给拒了，实在没有时间面这么多 company，又不想浪费别人的时间。不过心里还是觉得有点可惜，啧啧。

贴一下 M 的经历吧。On campus 就一轮，30 分钟。Interviewer 是个老中，一上来看我 resume，问为啥 phd 了还来面 sde。然后开始问 resume 上的东西，我借机会 sell 了一下自己。

Technical 问题只有两个：

1）Coding 题非常 old 了。两个 string 找最长 common substring。这个当场肯定 code 不了 subffix tree。于是就用暴力的方法，三下五除二搞定。然后问 complexity，如何改进，bla bla bla。竟然忘记了说可以用 DP，低级失误啊。不过面试官还算满意。

2）你认为 bing 有什么可以改进的（我投的是 bing）？你 research 做的东西有没有可以 apply 的？

然后时间够了，面试官说三周内给结果。一周之后就收到了 onsite 通知。攒人品，祝大家好运！

Pasted from <http://www.mitbbs.com/article/JobHunting/31451397_3.html

>

今天又做 coding 面试了，上来就要写个函数 返回二叉排序树的第 k 个最小的 node。
我写了一半，感觉不对劲。请大侠赐教。

Pasted from <http://www.mitbbs.com/article/JobHunting/31451705_3.html

>

you are given a M x N matrix with 0's and 1's
find the matrix with largest number of 1,

1. find the largest square matrix with 1's
2. Find the largest rectangular matrix with 1's

没想到很有效的办法。大家见集思广益~

Pasted from <http://www.mitbbs.com/article/JobHunting/31452521_3.html

>

有几个同学问面筋，不太记得起来，很多版上是有的，所以觉得那些面你的人水平挺一般的，下面贴两个印象深刻的：

1。这道题被好几个不同的公司面到过：Fibonacci 数列，一般让你给一个 recursive 的版本，然后写个 iterative 的版本，然后问有没有更快的可能性。我记得以前在某个版讨论过，参考 wiki:

这样的方法，可以在 O(log(N))的时间和 O(1)的空间复杂度内算好。要写程序的话，用类似下面的方法：

```
Matrix2x2 F[][2] = {{1, 1}, {1, 0}}, Fn[2][2] = {{1, 0}, {0, 1}};

while (N) {
   if (N & 1)
      mul(F, Fn, Fn);    // Fn = Fn x F;

   mul(F, F, F);         // F = F^2;
   N = N >> 1;
}
```

2。另外一题很简单，但是蛮 tricky 的。How to test if a number "a" is power of 2

```
return (a-1) & (a) == 0;
```

网上经常有问怎么样判断一个数里面有多少个 1 的位数，这个只是其中一个最简单的特

Pasted from <http://www.mitbbs.com/article/JobHunting/31452533_3.html

>

于完成了 F 公司历时 2 个月的所有 interview，总算可以松口气了，据称他们下周一开会讨论，希望最终会修成正果。来说点经历吧。

多亏好朋友 Z 帮忙 forward resume，很快就来了第一轮 phone interview。编程题还有点老：

[Coding Q1]: Given an array A, output another array B such that B[k]=product
of all elements in A but A[k]. You are not allowed to use division.

其实这题 interview 之前在本版 JHQ 看过，可是当时看的题目太多，没有去想 solution。所以刚开始听到这题还 surprise 了一下。我觉得这个不能用除法的限制太无聊了（建议改个 problem 来问这个 algorithm），于是忍不住问 why not division，顺便拖延一下时间想算法。面试官说除法慢...显然不是什么很 convincing 的理由，我说那乘法也慢啊。说完我已经想到怎样做了，于是顺利过关。

接着就来了比较衰的第二轮，题目是这样的：

[Coding Q2]: You are given a string e.g."face" and a set of mutation rules,
e.g. a->@, e->3, e-E. Print all the possible strings that can be generated b
y the rules, e.g. f@c3, fac3, etc.

其实就是 BFS 再加上 hash table 判断是否重复 print。马上就想到 algorithm，面试官说好，你开始写吧。然后问题就来了，太久没写 c++忘了 hash table 的函数定义。好像依稀记得 hash table 还有几个版本，想了一会没想起来，又不好意思问，汗！最后还是忍不住问了，他说你随便给个函数名和接口吧。最后磕磕碰碰总算把程序写完了，却给人留下了很不好的印象，感觉写程序很不熟！据说最后这个人给了我一个 borderline，还算好，没把我 fail 掉。真惭愧啊，可怜我还是写 c++起家的...

因为第二轮不太理想，本来应该两轮过后就 onsite，结果 hr 来信说要第三轮 phone，还很好人的说 We do have three phone interviews at times. We are constantly evalua ting our process so I apologize for the change. 第三个面试官又临时换人，最后居然是同系师弟，不过之前没见过面，不然可以套近乎了。。。言归正传，换人大概还是因为他们组想看看我 match 与否。他问的都是 machine learning，风格和原来完全不同，还多多少少有点 surprising 的。

[Coding Q3]: Implement one step of decision tree which splits the node into two subtrees.

之后还讨论了一些 learning 的问题，我问了他们用的 technique，有什么存在问题等等，相谈甚欢。

关于 onsite，因为签了 NDA，不方便透露题目。请大家也不用发信来问了，做人还是要讲信用的。只看面经的各位看官可以略过以下了。。。至于那个 onsite 可谓一波三折。本来订了机票周四晚上到 sfo，周五中午 onsite，挺好的 schedule。结果某 airline 居然机件故障，把飞机拖去修了几个小时，又不肯调其他飞机来，白白 miss 掉了从 vegas 飞 sfo 最后一班航班，被迫在 vegas 住了一晚。下了飞机都晚上 12 点了，随便找个 airport 旁边的 hotel 住下，改了第二天最早一班机。结果第二天又晚点两个小时！据称 SF 大雾，traffic control...折腾了半天到了 F 都周五下午两点多了，又累又紧张。连 HR 也只好说：it's hard to visit us...不过不管那么多，灌了杯 coffee 就上阵了，结果还好，没有想象中 intensive，也发挥自己的水平了. 因为 onsite 去的太迟，没见到 manager，HR 又说 schedule TWO more follow-up，其中一个是 manager。OMG，我说好，那就 back-to-back 吧。上周终于面完真是 relieved 啊，前后 5 轮，历时 2 个月。

关于面试的经验教训，我的感觉是 1）F 的 interview 是比较严谨的，phone interview 就要 candidate 在 white board 上把 code 写出来，不是说说算法就算了，detail 也问得很仔细。因为他们要求员工 follow 整个 project，从 idea，到 algorithm，到 implementation，而不是自己想个东西出来让别人写 code 实现就完了。各位像我一样平时写 research matlab code 多于写 c++的 phd 要注意多练练手了，小心阴沟翻船。2）Never give up 无论 interview 多么不顺利。不要被外界的不利因素 distract 自己，该准备什么就好好准备，我相信 life has miracles.

PS: 貌似 F 的同学们也会上来 job hunting 版。文中若有冒犯之处，请多多包涵。F 的 S 同学，你那题大概说了也不要紧吧，没理解背下来也没啥用的。F 的 Y 同学，我没有泄漏你的面试题，以后还可以继续用，哈哈。

Pasted from <http://www.mitbbs.com/article/JobHunting/31452725_3.html

>

看了一晚上精华区，发现这道题有问题啊。

5。Given a graph (any type - Directed acyclic graph or undirected graphs
with loops), find a minimal set of vertices which affect all the edges of
the graph.
An edge is affected if the edge is either originating or terminating from
that vertex.
The time should be less Q(n^2)

这个题就是最小顶点覆盖问题吧？

或者是我对最小顶点覆盖问题理解有误？或者对这题理解有误？

Pasted from <http://www.mitbbs.com/article/JobHunting/31452961_3.html

>

00*100 部分有序矩阵数组的排序

有 100 个有序数组(从小到大)，每个里面有 100 个数。

设计一个算法合并这个一百个有序数组，中间步骤只允许多申请一个大小为 100 个数的空间（也就是一个数组的大小）。

Pasted from <http://www.mitbbs.com/article/JobHunting/31453089_3.html

>

1. How to call C++ code in C? How to call C code in C++?
2. In which three cases, initialization list has to or is preferred to be used for a constructor?
3. Can we design singleton by setting all the data member and method of a class to be static?

Some others:
4. Is overloading allowed in C? If not, how to differentiate them?
5. Default methods that are generated by a class in C++.
6. Difference of struct and class in C++
7. Given a class has first name, last name, SSN and etc.Need to query according to first name, first name + last name, what STL should be used? (If map/multimap, what should be the key?) How the query should be? How to query all the first name that initialed as "J"?
8. Meaning of static in C and C++
9. Meaning of inline in C++; where should it be used?

几天上午一个面试的问题。有些东西没用过，虽然以前看过，但是还是没有答出来；都去准备其它的去了，没想到全是问 c/c++的问题。还是有些不服气，move on。

Pasted from <http://www.mitbbs.com/article/JobHunting/31454759_3.html

>

形象投票:
形象得分: 291 分

我的博客

0 0
[上篇] [下篇] [同主题上篇] [同主题下篇]

发信人: mandman (满老), 信区: JobHunting
标 题: [合集] 一个 Google / MS 面题

发信站: BBS 未名空间站 (Wed Oct 28 23:41:26 2009, 美东)

☆──────────────────────────────────────☆

wisher1 (wisher) 于 (Fri Aug 7 16:59:50 2009, 美东) 提到:

在精华区里看到多次, 都没有答案.
删除一个 singly linked list 节点, 但不知道 head.
不知道 head, 怎么找之前的那个节点阿? 又没说有 loop.

谁知道 trick 在哪?

多谢!

☆──────────────────────────────────────☆

chaicar (chaicar) 于 (Fri Aug 7 17:05:15 2009, 美东) 提到:

record the next node of it
delete its next node
do the assignment to copy saved next node to it

Pasted from <http://www.mitbbs.com/article/JobHunting/31454761_3.html

>

Write code for finding number of zeros in n!
OR
Find the first non-zero digit from the right in 100! (Factorial of hundred).
Can an int store hundred factorial. What size of array should be sufficient
to solve the above problem. Write a code for the same.
大家有什么 idea?

Pasted from <http://www.mitbbs.com/article/JobHunting/31454839_3.html

>

you have a billion google searches a day, design a data structure which
lets you pull out the top 100 unique ones at the end of the day.

我的想法是 create hashtable

scan billion data 一次，在 hashtable 纪录每个 query 的次数

然后再 scan billion data 一次,通过 heap 和 hashtable 找到 top 100

不过这样的话,billion data 会被 scan 2 次，disk i/o 会很大

不知道有没有什么 scan billion data 一次就可以找到 top 100 的办法

大家讨论一下

Pasted from <http://www.mitbbs.com/article/JobHunting/31455781_3.html

>

er executing this code:

```
list<int> L;
...
list<int>::iterator in_range =
    find_if(L.begin(), L.end(),
           compose2(logical_and<bool>(),
            bind2nd(greater_equal<int>(), 1),
            bind2nd(less_equal<int>(), 10)));
```

What is the best assertion that should be used as a post-condition?

```
assert(in_range == L.begin() || (*in_range >= 1 && *in_range <= 10));

assert(in_range == L.end() || (*in_range >= 1 || *in_range <= 10));

assert(*in_range >= 1 && *in_range <= 10);

assert(in_range == L.end() || (*in_range >= 1 && *in_range <= 10));

assert(in_range == L.end() && (*in_range >= 1 && *in_range <= 10));
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31456679_3.html

>

he difference of following two expressions: Test A or Test B()

```
Class Test;
Test A;
or
Test B();
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31456597_3.html

>

When a derived class is destructed, at what stage will the base class's destructor be called?

the answer varies on a case-by-case basis

It will automatically be called after the destructors for the derived class
data members

It will automatically be called before the destructors for the derived class
data members

It should explicitly be called at the beginning of the derived class
destructor

It should explicitly be called at the end of the derived class destructor

Pasted from <http://www.mitbbs.com/article/JobHunting/31456485_3.html

>

什么样的情况下用 virtual deconstructor？
2. virtual function 是如何工作的？virtual table 是如何实现的？
3. virtual function 具体调用哪个 function 是在编译的时候，还是在代码执行的时候
决定的？
4. 类的 copy constructor 和 assignment operator "=" 有什么区别？有什么主意事
项？

我的概念不是太清楚，请好心人帮忙回答。有包子送。

什么网站有 C++的测试题库，哪有 free 的可以练练手？bloomberg 考的 C++问题哪里能得
到？
谢谢！

Pasted from <http://www.mitbbs.com/article/JobHunting/31457805_3.html

>

给定一个二叉树的一个 node，编程返回中序遍历的下一个 node。如果最后一个，返回 null, 怎么做？

Pasted from <http://www.mitbbs.com/article/JobHunting/31459733_3.html

>

网上看来的：

一堆数，其中一些数出现了一次，一些数出现了两次，只有一个数出现了三次

找出那个出现了 3 次的数

hash 方法很 trivial 就不说了。

如果用 bitwise operator，怎么高效的做？除了 XOR，是不是还得用点别的办法？

Pasted from <http://www.mitbbs.com/article/JobHunting/31460327_3.html

>

就是一直一篇文章，球可以覆盖所有单词的最小窗口，记得 bbs 有几个人提过这个题，但是没人给过解，希望牛人能够赐教！！感激！

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31460569.html

>

1. If the probability of rain tomorrow is twice than no rain.
What is the probability of rain tomorrow:

2. A grass, 3 cow can eat 3 days. 2 cow can eat 6 days. How long can one cow finish the grass?

Pasted from <http://www.mitbbs.com/article/JobHunting/31461095_3.html

>

Given a document and a query of K words, how do u find the smallest window that covers all the words at least once in that document? (given you know the inverted lists of all K words, that is, for each word, you have a list of all its occurrrences). This one is really hard. Could someone propose an algorithm in O(n)?

Pasted from <http://www.mitbbs.com/article/JobHunting/31461767_3.html

>

Desgin an algorithm to find whether a given sting is formed by the Intealeaving of two given strings. 注意，原来的两个 given strings 的本身的 character 的顺序不能变。

这个题不简单，因为你不能简单的用 3 个指针分别指向三个 string,遇到 string A 的就拷贝到 dst string,遇到 string B 的就拷贝他的。最麻烦的在于遇到 A,B 都相同的，你不能 advance both ptrs until they are different and then move one of them back.
The point is who is to be moved back? You cannot simply randomly choose one.
For example,
stringA: ABCEF...
string B: ABCA...

dst string : ABCABCEF....
那么，如果取 B's ABCA 就错了。

哪位大侠能指教怎么做呢？

Pasted from <http://www.mitbbs.com/article/JobHunting/31463527_3.html

>

Given a set of points (x,y) , find all pairs of points whose distance is
less than a given number, say, K.
这个题 brute force： 对每个点，求和其他点距离，O(N^2),不知道哪位大侠有高见啊
？请不吝赐教！万分感激！

Pasted from <http://www.mitbbs.com/article/JobHunting/31463131_3.html

>

{1， 5， -5， -8，2， -1，15 }
要把负的扫到左边，正的扫到后边。
不能改变顺序
得到{-5 -8 -1 1 5 2 15}

这个题有 time 低于 n^2 space=O（1）的解法吗

Pasted from <http://www.mitbbs.com/article/JobHunting/31464055_3.html

>

这些东西我很都不熟悉。希望有高手指点指点，呵呵

1. Mempool design with 30k limit.
mempool 是应该在一开始就 allocate 30k 连续的内存,然后分配和管理？

或者是每次 call allocate(n)的时候再通过 operator new[]来分配内存，update size member？如果是的话，free(ptr, n)怎么写呢？貌似 operator delete[]不能带 size 参数啊？

总之我就是对 memory design 这块很不熟悉。。。

2. Implement put/get methods of a fixed size cache with LRU replacement algorithm.
这个是不是用 fixed size 的 max heap 来实现？每个元素定义一个 key，表示距离上次使用的时间，每使用一个元素，就相当于是把它的 key 更新为比当前最小值更小的数，然后做 heapify()操作？
每 put 一个元素，就 assign 新元素一个最小的 key，然后用新元素替换掉堆顶点，然后做 heapify?

3. Write a function to implement a buffer for DataOutputStream.
这个我完全没啥概念了。。。求指点一下~

4a. How do you write malloc and free to detect memory reference violation?
4b. flag a block of memory as used by putting some bit pattern at the beginning of the block. What bit pattern will you use?
这两个问题也是摸不到头脑的。。。

5. How to implement singleton without using static/global variable?
完全没思路，design pattern 我基本上就是临时抱佛脚都还没抱上。。。

Pasted from <http://www.mitbbs.com/article/JobHunting/31464509_3.html

>

其实这些题也适合别的 OS,只不过面试的这个职位是基于 Linux 的。

1. buffer overflow 的工作原理
问的特定环境是： 在 client-server 的 model 下，client 是如何通过网络造成 server 上
的 buffer overflow,从而在 server 上制造出 security hole?

俺只知道，肯定是 client 给 server 发的 packet 中，故意把特定的 field(比如长度)弄错
，使得 server 上的程序在 copy 的时候，造成 buffer overflow (因为一个特大的 length),
谁能说说到底 buffer overflow 是如何产生的？有什么好的文章，或者网站 link 介绍这
个问题的？觉得搞 network security 的同学应该很明白。
谢谢！

2. 关于 TCP 的实现的问题（1）
操作系统中 TCP 的实现用到了几个 timer,分别是什么？
这个题怪怪的，谁能知道这样的细节？

3.关于 TCP 的实现的问题（2）
TCP packet header 中的 Window size (接受方的 window size)的 update 是多久进行
一次？就是问接收方在什么情况下，或者是多么频繁向发送方 update 新的 windows size?

4. 关于 socket
TCP connection 用 socket 建立后，有可能有很长时间通讯的双方没有任何数据来往，
比如 telnet client 登录 telnet server 后，可能人会离开很长的时间，这个时候 TCP
server 怎么知道 TCP client 是 alive 还是 crashed?

如果你设计一个自己的应用程序，你该如何处理？就是问在你自己的 client 和 server 建
立 connection 后，你是如何 check whether the socket is still alive or not? 是在
你自己的应用程序中定时的发一些类似于"Hello" 的 packet 作为查询呢？还是 OS 的
socket 能够自动的提供 the information about the socket status?

5.如何用 C 语言实现 object oriented programming?

6. 关于 kernel synchonization

在 SMP 系统下，用 spinlock,还是用 semaphore 来作 synchronization 比较好？为什么？

俺只知道如果你的代码不能 sleep 的时候必须用 spinlock,比如在 interrup handler 里面
。还有就是如果用了 spinlock,你要能够处理的很快。别的就想不出有什么区别了？

不过 Jonathan Corbet 的"Linux device drivers"一书中说在 Linux kernel 的实现中，
spinlock 引入的主要目的是为了让 Linux 在 SMP 系统里运行的更有效，不知这是为什么？

俺被问的一塌糊涂，很郁闷，请大拿们给启发一下。
谢谢！

Pasted from <http://www.mitbbs.com/article/JobHunting/31466547_3.html

>

如题，职位是 web engineer,希望有人可以用到。
第一次电话是 recruiter 的，按清单问了些问题：
1. say some http methods?
2. get/put difference?
3. what does DTD for xml mean?
4. common protocol used in layer 4?
5. describe different ways to use css in html

?
6. difference between well-formed and valid xml?

前两天第二轮 technical phone interview：
1. why and how did u get into web development?
2. what do u like about web development? not like about it?
3. why do u want to work for google? 我扯到 ajax 的推广，他顺着问 ajax principle, security issue
4. what language are you comfortable with? talk about it. why and how did people design it?
5. explain 3 components of MVC
6. what happens when a user types google.com in URL bar and press enter? (dns, http get, tcp connection establishment, etc)
7. what may slow down the rendering of html

page when its contents have been downloaded from server? (load other resources like css,js and parse them, etc)
8. read n lines of random numbers(space as delimiter) from a file, lines with same numbers are treated as duplicated lines, regardless of the order. check and print non-duplicate lines. performance time analysis.

顺带问一个转身份的问题：如果我 H1->F1->H1，重新转回 H1 的申请被拒了，那是不是还停留在正常的 F1？ 同样，其它转身份，例如 F1<->F2, F2<->H4 互转之类，如果申请被拒，是正常停留在之前的身份吗？ 前几天看到个帖子，H1 transfer 被拒，身份就黑了，不知道跟这些例子有什么区别，谢谢。

Pasted from <http://www.mitbbs.com/article/JobHunting/31467259_3.html

>

1. there are only 6 db connections in the pool, all 6 are being used,
another request needs to connect to DB, it does not want to wait. How to do it? One solution is to make a new connection and add it to the pool. But the interviewer wants standby solution. Anyone knows the standby solution?
Thanks

2. For the db connection. min = 10 and max = 40.
Will 10 connections be created at server start up time?

2.1 If we start with using 13 connections, when all the jobs are done, how many will be kept in the pool? 10 or 13?

2.2 If we start with using 45 connections, when all the jobs are done, how many will be kept in the pool? 40 or 45?

Pasted from <http://www.mitbbs.com/article/JobHunting/31467451_3.html

>

uppose there are n cities, and there may / may not be flight route
between c1 to c2. Design data structure to store this information and write a function that receives two cities name, and return whether or not there is a flight between them (either directly or through connections)

Pasted from <http://www.mitbbs.com/article/JobHunting/31469019_3.html

>

不 trivial

Given a 3x3 square:

1 2 3
4 5 6
7 8 9

You are allowed to do circular shift on any row, and circular shift on any column, as many times as you please. Question: can you switch position of 1 and 2 with the allowed circular shifts?

Pasted from <http://www.mitbbs.com/article/JobHunting/31469459_3.html

>

通常看到这种题目都感觉有点头疼。比如，design a messaging system. an online poker room.大家说说看

Pasted from <http://www.mitbbs.com/article/JobHunting/31470087_3.html

>

刚刚 on-stie 面试完某大公司。面试了 7 个人，大概问了 20-30 道题，有 1 道题不会，尽管其他的都打上来了，很是郁闷，本以为自己准备的足够好了，哎。但是这道题不会，很不甘心，希望大侠们帮助！！！

In our indexes, we have millions of URLs each of which has a link to the page content, now, suppose a user type a query with wild cards *, which represent 0 or multiple occcurrences of any characters, how to build the index such that such a type of query can be executed efficiently and the contents of all correpsonding URLs can be displayed to the users? For example, given a query http://www.*o*ve*ou.com. You man need to find iloveyou.com, itveabcu.com, etc.

以前我见过类似用 wild card 来做 query 的，就*来说，一个方法是用*split query into
a few parts, for example, *o*ve*ou => o, ve, ou, 然后分别用 o, ve, ou 查询，
但是似乎不适合这道题。 另外，如果对 Index 里的每一个 URL 建 suffix tree ，然后对
每个 query check againgt 所有的 suffix tree, 这样实际上就是 scan all urls, 明显
也不合适。但是排序？我想不出来。

这道题真难倒我了。。。。。。求求大侠赐教。不仅仅是为了面试，就是觉得不甘心。
。。多谢！！！！

Pasted from <http://www.mitbbs.com/article/JobHunting/31472965_3.html

>

ftware Development Engineer

问候之后，首先问了一下我的 research, 让我具体的阐述我提到过的算法

然后 OOD 的一道题，其实不难，但我感觉自己答得不好
have a furniture class, some child classes like table, chair, etc.
they want to extend the class hierarchy, as there are wood table, steel
table, wood chair, steel chair, and so on.

我首先给出 class + interface 的 design:
furniture (table, chair, ...)
table (woodtable extends table implements wood, steeltable extends table
implements steel)
chair (similar as table)
然而 interviewer 立刻指出这样的话，如果要加 fire, 或者和重量有关的 functions 时，
会有 code duplication (因为 wood 可燃，steel 不可燃, assuming fire function is
defined in wood and steel interfaces). 我最后说那把 wood 和 steel 也改成 class (c
++ multi-inheritance), 这样可以 avoid code duplication. Interviewer 又问那么如
何实现 woodsteeltable? 我说就 inherit woodtable and steeltable. 自己对这个问题
总体感觉不好。大家有什么好的 design?

后面的问题比较简单：

given a deck of cards, how to shuffle it?

having a web application, front-end, middle layer and database. How to scale database to accommodate increasing traffic?

Pasted from <http://www.mitbbs.com/article/JobHunting/31471911_3.html

>

onsider a function which, for a given whole number n, returns the number of ones required when writing out all numbers between 0 and n.

For example, f(13)=6. Notice that f(1)=1. What is the next largest n such that f(n)=n?

Pasted from <http://www.mitbbs.com/article/JobHunting/31471823_3.html

>

请解释 garbage collection？
在 garbage collection 中，对 circular reference 的你怎么办？

Pasted from <http://www.mitbbs.com/article/JobHunting/31470831_3.html

>

以下文字转载自 SanFrancisco 讨论区 】
发信人: soldiercrab (老军医专治装 B 文学小青年), 信区: SanFrancisco
标 题: 一朋友被 Google 的电面干掉了
发信站: BBS 未名空间站 (Sat Dec 5 15:03:55 2009, 美东)

栽在一道编程题上：Find a longest increasing subsequence in an integer array。

问问题的人要求朋友拿出 O(nlog(n))的算法，但朋友只给出了 O(n^2)的 dynamic programming 的方法。其实我觉得给出 dynamic programming 算法足够进入下一轮了。那个 O(nlog(n))的算法好歹也值当年一篇 paper，而且貌似不是那么直观。电面就想出来不容易。不过多半是我段位不够，还不够 Google 的要求。或者朋友的 dynamic programming 其实错了（这道题要倒过来找，稍微绕一点点）。

Pasted from <http://www.mitbbs.com/article/JobHunting/31473303_3.html

>

发信人: MeetYouThere (Meet You There), 信区: JobHunting
标 题: yahoo 面试题
发信站: BBS 未名空间站 (Sun Dec 6 18:26:44 2009, 美东)

Given n points in the form (x1, y1, z1)…..(xn, yn, zn), find the k closest
points to the origin.

Given the same points as above, find the K closest points to each other.

有什么快速解法没

--

※ 来源:•BBS 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM:
71.198.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31473965_3.html

>

1. one array filled with numbers from 1 to N, but one number is missing. wha
t's the most efficient way to find the missing item? what about two or more
numbers are missed?

2. find the repetative chars in a string and delete them

3. find the binary tree from its preorder and inorder traversal

Pasted from <http://www.mitbbs.com/article/JobHunting/31474331_3.html

>

and(5) generates a random integer number between [1, 5], how do you
generate a random integer number between [1, 7] when you can only call
rand(5)?

Pasted from <http://www.mitbbs.com/article/JobHunting/31476251_3.html

>

鉴于安静的气氛，来一个：
Given an integer, print the closest number to it that is a palindrome
input: 1224

return: 1221.

Pasted from <http://www.mitbbs.com/article/JobHunting/31477969_3.html

>

Given a value and a binary search tree.

Print all the paths(if there exists more than one) which sum up to that valu
e. It can be any path in the tree. It doesn't have to be from the root.
我理解是这个 path 可以是其中任意一截,不用包括头尾

Pasted from <http://www.mitbbs.com/article/JobHunting/31478003_3.html

>

第一个面试官是个中国人，女的。
开始想同她套套近乎，也不知道是不是套错了，反正当时感觉她不喜欢我（其实后来回头想想，可能不是她不喜欢我，她就是那样说话的态度，是我感觉不对）。
编程题
given a character string, print the number of occurence of each charcater in
order. ie. if the string is "ceabcw", then you should print something like:
a 1 b 1 c2 e 1 w 1.
she asked the possible data strucutre to approach. I gave array, hashtable,
and BST. she asked me to use BST, and using no recursive. Also how to handle
unicode.
然后问了一些测试题，让我测试她们的一个产品。细节忘了，总之她对我不满意。我也觉得基本没戏了。
第二个是个印度人。编程题：
given a matrix(assume it is a bitmap), print all cells that is on.
做的不好。后来问了一些测试题。

第三个是个白人。
开始问测试的问题，回答得一般。因为觉得已经没有戏了，所以也不大有精神。
编程题很简单，是实现阶乘。不过有个问题没有考虑到，就是 overflow 怎么处理。

总之非常惨，第一次面试这么惨。也请给位说说自己的想法怎样解答那些问题

Pasted from <http://www.mitbbs.com/article/JobHunting/31481069_3.html

>

两个玩家， 一堆石头，假设多于 100 块, 两人依次拿， 最后拿光者赢， 规则是

1. 第一个人不能一次拿光所有的
2. 第一次拿了之后， 每人每次最多只能拿对方前一次拿的数目的两倍

求先拿者必胜策略, 如果有的话

Pasted from <http://www.mitbbs.com/article/JobHunting/31482015_3.html

>

题目：

从一个 string 变到另一个，比如"study"->"world" （字数相等），要求

1. 每次变一个字母
2. 每次改变后的 string 必须是一个词典里面能查到的英语单词，比如你不能把 study 变成 atudy

Pasted from <http://www.mitbbs.com/article/JobHunting/31482527_3.html

>

google interview question from glassdoor

Design and describe a system/application that will most efficiently produce
a report of the top 1 million Google search requests. You are given:

You are given 12 servers to work with. They are all dual-processor machines
with 4Gb of RAM, 4x400GB hard drives and networked together.(Basically,
nothing more than high-end PC's)
The log data has already been cleaned for you. It consists of 100 Billion
log lines, broken down into 12 320 GB files of 40-byte search terms per line.
You can use only custom written applications or available free open-source
software.

Pasted from <http://www.mitbbs.com/article/JobHunting/31483445_3.html

>

u are given a binary search tree,
each node has a parent, left and right
do pre-order/in-order traversal without stack.
cannot change the structure of Node.

test cases: 8 6 7 5 4 9 10 11 12

test your codes using the test case above.

Pasted from <http://www.mitbbs.com/article/JobHunting/31483789_3.html

>

关于排列组合的程序问题, 我一只都没理解太清楚, 现在厚脸皮来请教一下. 这些问题一般都要涉及到递归, 我这里不是问的算法的问题, 而是程序的实现问题. 我一直不知道怎么实现才是对的.

比如, 5 选 3 的全组合, a,b,c,d,e.

1 中间结果怎么保存, 是用一个 vector 来保存,还是用多个 vector 来保存?
2 如果用一个 vector 来保存, 递归的时候, 最终状态是什么? 何时 pop, 何时 push, ?

请问有谁可以贴个 code 学习一下么?

Pasted from <http://www.mitbbs.com/article/JobHunting/31484637_3.html

>

Given an array, find the longest subarray which the sum of the subarray less
or equal then the given MaxSum.
int[] FindMaxSumArray(int[] array, int maxsum)
for example, given array: {1, -2, 4, 5, -2, 6, 7}
maxsum=7
the result would be: {1,-2, 4, -2, 6}

Pasted from <http://www.mitbbs.com/article/JobHunting/31484653_3.html

>

iven a integer, output its previous and next neighbor number which has
the same number of bit 1 in their binary representation.

下面为什么去判断(number & 3) != 2？

```
while ((number & 3) != 2) { // for right neighbor, change this line to
// (number & 3) != 1
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31485091_3.html

>

要求当场写 code。

1 下面的 int * takeaddress()有没有问题， 啥问题？

2 写个效率高的 takeaddress 出来， 实现同样的功能

```
int  * paddress, address1, *r;
paddress = takeaddress(); /* defined below */
address1= paddress[0];


int * takeaddress()
{
   int  address[8];
/* The address are defined here */
  return  address;
}
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31485465_3.html

>

一个字符串，要求返回重复次数最多且最长的子字符串（假设源字符串中最长重复次数最多的子字符串只有一个）。例如 “abcabcdfabcdf”要求返回“abcdf”. 因为“
abcdf”重复次数最多且最长。

俺只想到两个土办法：
1)找到所有字符串组合（例如 a, ab, abc, abca, b, bc, ...），都放入 hash table
，找重复次数最多的且最长的。
2)用 Dynamic Programming 找 LCS 的办法，两个字符串都是源字符串，然后在那个 2D array 里面找最长 match，并计算重复的次数，然后输出结果。

感觉两个方法的 time complexity 都挺大的，不知大家有没有什么别的想法？

Pasted from <http://www.mitbbs.com/article/JobHunting/31485529_3.html

>

很多都是老题，不过我专门整理了一下：
1. string match:
string Text, Pattern;
find a substring of Text matches with Pattern.
解法纲要：Rabin-Karp, KMP, suffix tree

变种 1b: multiple match:
string Text, PatternSet[n];
find a substring of Text matches with any one pattern in the set;
解法纲要: Rabin-Karp

2.LCSubstring:
string A,B;
find the longest common consecutive substring;
解法纲要：DP(A.len*B.len 复杂度)，suffix tree(A.len+B.len 复杂度)

3.Longest Palindrome
string A;
find the longest substring of A which is a palindrome;
解法纲要：类似 2

4.Wild card match:
4a: Pattern contains '?'(s)
4b: Pattern contains '*'(s)
4c: Pattern contains both;

//以下是与 dictionary 有关的题目

5. dictionary + wild card search(一般都需要做适当预处理):
第一种 search：search 所有 match 结果
第二种：返回某个特定的结果，比如，所有 match 中最长的单词
5a: pattern = ??a????b* (指定某些位上的字母)
5b: pattern = abcde* (指定 fixed/unfixed length 的前缀)
5c: pattern = ?a*bcd*e?f* (?和*任意混合搜索)
解法：待探讨

6. dictionary + 包含字符集合:
Letter_Set = "aabbbcd";
第一种 search: 所有至少包含 2 个 a,3 个 b,1 个 c,1 个 d 的单词
第二种 search：所有至少包含这个字母集合的单词中最长的/最短的
解法：待探讨

7. convert a valid word to another valid word of the same length, by
replacing one letter in one step, every intermediate word must also be valid;
解题思路：相同长度的单词构建一个图 + BFS

8. edit distance (misspell correction):
type a misppell word, give top10/all suggestions of correct words;
解题思路：首先定义计算 edit distance 的 metrics，然后从每个 valid 单词计算出到它距离<=某给定值的所有 misspell 的单词(类似 BFS 的一层一层的算)

9. find a matrix with max area: each row and each column of the matrix must
be a valid word;

10. 朴素搜索，在 dictionary 中搜索一个单词是否存在：
解题思路：hash; trie;

10b. shortest unique prefix: give a string, find its shortest prefix, which
doesn't match with any prefix of any valid word in dictionary;
for example:
cat against {dog, be, cut} is ca
cat against {dog, be, cut, car} is cat
cat against {dog, be, cut, car, cat} is null
解题思路：trie/prefix tree;

11. solve a crossword puzzle;

Pasted from <http://www.mitbbs.com/article/JobHunting/31485923_3.html

>

1. N 台机器，每台机器有 N 个数
找 median (2 个数组找 median 的扩展版)

2. 已知 coin denominator set，例如,2cent, 3cent, 5cent...
给定一个目标数，比如 126cents
最少需要多少个 coin。

这个题我以前问过一次，没人回。。。我觉得是很好的题，贪心，回溯，DP 都可以试试
。但是我一直没找到最满意的解。

3. 一个整数数组，找 3 个数满足勾股定理。求比 O(n^2)更好的解

Pasted from <http://www.mitbbs.com/article/JobHunting/31486805_3.html

>

glassdoor 上看到一道题目：

Given a file of unknown size, devise an algorithm to give equal probability randomization to choosing a single line given a one line buffer space.

请教思路？

Pasted from <http://www.mitbbs.com/article/JobHunting/31487119_3.html

>

1. 很长的 log file 记录了用户访问 amazon.com 的过程，两列分别为 userID 和 pageName.
log 从上倒下按照点击发生的时间顺序。找出最 popular 的 3 连击。
eg:

```
zhang  welcome
Li     Hello
Wang   welcome
Li     books
Wang   Hello
zhang  books
Li     shopping cart
Li     checkout
zhang  shopping cart
Wang   camera
zhang  checkout
```

最 popular 的 3 combo 是 books -> shopping cart -> checkout

2. Permutation of a string.这题最郁闷，我把 programming expose 里的 code 默写了出来。但这个方法是不管字符重复的，假设都是不同的。现在考官要不显示重复的，而且他要求不能先
都列出来再剔除，而要在发现重复的时候及时制止。没想出来

3. Design a fight ticket booking system.

4. 老板说网站很慢怎么办？

老板说数据库很慢怎么办？

没一个很清楚的，每题都说了点。高人说说这几题标准都怎么答啊？谢了！

Pasted from <http://www.mitbbs.com/article/JobHunting/31487345_3.html

>

不好意思，刚才不小心把第一个版本删除了

上来好几个 behavior 问题，一般 behavior 问题我不怎么准备，答的很简略，都是常见的问题，我就不列了。

c++ and data structure
** single linked list, find nth from the end
** Overwriting and Overloading
** Stack vs. Queue
** Array of integers, all integers appear even times except one, find the one appears odd times. (some following up questions for this one)

OOD
** Do you approve the following design?
```
Class Furniture{
    Some functions related to the property of furnitures;
};
4 classes derived from Furniture
Class wood_chair
Class steel_chair
Class wood_table
Class steel_table
```

What if you need to design a lot of other furnitures like desks.... with other materials like plastics

**An open question.Takes more time than any other questions. It is related to the project they are working on, you should not be asked.

Good Luck to All of us :D

Pasted from <http://www.mitbbs.com/article/JobHunting/31487819_3.html

>

1. given n strings with equal length, say x. find the substring shared by
all of them. For example, abcx, abdx, abea, then ab is shared by all of them.
2. the gmail page loads very slow. any suggestion for improvement?
3. we want to check the number of querys obtained from the world in the last
minute and the last hour, what data structure should you use for that? If
there are billions of records, i.e, too many records for the main memory,
what suggestions do you have?

马上就要第二轮店面了，求大家的题目和建议，我们也可以私下交流。谢谢!

Pasted from <http://www.mitbbs.com/article/JobHunting/31487921_3.html

>

你有一种语言的 dictionary,你有一大串 string,没有 delimit,你如何 interpret 成字典中的字呢？

Pasted from <http://www.mitbbs.com/article/JobHunting/31488093_3.html

>

Given a binary tree

```
struct node{
struct node* leftChild;
struct node* rightChild;
struct node* nextRight;
}
```

The nextRight points to the right node to the current node in the same
level. Ask you populate the nextRight pointers in each node.

大家怎么做这道题.

Pasted from <http://www.mitbbs.com/article/JobHunting/31491521_3.html

>

今天 facebook 第一面，现在 hr 都开始问技术问题了。。。问我会什么语言，我就说 C++ best, 她就
问我一些很基本 C++的问题，还有两个 bubble sort best case 的复杂度和一个排序的
思路。

最后一题听的不太明白，题目剧长，头都大了。

面完之后给了 puzzle 的 link，要求做 meal 和 buffet 里挑一个。

问下大家哦，是不是 Puzzle 之后才是 phone screen 啊？大家 puzzle 都是自己做的吗？或者哪里有前
辈留下的 hint 吗？

Pasted from <http://www.mitbbs.com/article/JobHunting/31494081_3.html

>

1。Java 里如何比较两个 objects 是否相等
2。怎样找出一个 list 是否包含循环
3。inheritance 和 composition：什么时候需要用到哪种？
4。一个 int array
如何找出 subarray，使得元素之和最大
比如{-2,3,-1,3,-4}
那么答案应该是{3,-1,3}

Pasted from <http://www.mitbbs.com/article/JobHunting/31494489_3.html

>

发信人: skydoor (海阔天空), 信区: JobHunting
标 题: amazon 的那道题目
发信站: BBS 未名空间站 (Sat Jan 16 21:41:59 2010, 美东)

看了网上的讨论, 关于 amazon 那道 wood steel table chair furniture 的题目

自己写了一个, 牛人帮我看看我是否写的有问题......

在我机器上编译通过, 运行成功.

```
#include <iostream>
using namespace std;

class stuff {
  public:
    stuff() {}
    virtual ~stuff() {}
```

```
      virtual void info() = 0 ;
};

class table : public stuff {
   public:
      table() {}
      ~table() {}
      void info() { cout<<"Table "<<endl;}
};

class chair : public stuff {
   public:
      chair(){}
      ~chair(){}
      void info() {cout<<"Chair"<<endl;}
};


class wood: public  stuff {
   public:
      wood(stuff * s): stf(s) { }
      void info() {cout<<"Wood "; stf->info();}
   private:
      stuff* stf;
} ;

class steel : public stuff {
   public:
      steel(stuff *s) : stf(s) {}
      void info() {cout<<"Steel "; stf->info();}
   private:
      stuff* stf;

};



int main() {

   stuff * wood_chair = new wood(new chair);
   stuff * wood_table = new wood(new table);
   stuff * steel_chair = new steel(new chair);
```

```
    stuff * steel_table = new steel(new table);

    stuff * wood_steel_chair = new wood(new steel(new chair));

    wood_chair->info();
    wood_table->info();
    steel_chair->info();
    steel_table->info();

    wood_steel_chair->info();

    delete wood_chair;
    delete wood_table;
    delete steel_chair;
    delete steel_stable;
    delete wood_steel_chair;


}
```

output:

```
Wood Chair
Wood Table
Steel Chair
Steel Table
Wood Steel Chair
```

如果需要 plastic, 只需要再从 stuff inherit 一个 plastic 类就行了

=========================另一种方案==================================

```
#include <iostream>
using namespace std;

class material {
```

```
public:
material() {}
virtual ~material() {}
virtual void info() = 0 ;
};

class wood : public material{
public:
void info() {cout<<"Wood ";}

};

class steel: public material {
public:
void info() {cout<<"Steel ";}

};

class furniture {
public:
furniture() {};
void setMaterial(material *m) { this->m = m;}
virtual ~furniture() {};
virtual void info() = 0 ;
protected:
material * m;
};

class table : public furniture {
public:
table() {};
void info() { m->info(); cout<<" Table"<<endl;}

};


class chair : public furniture {
public:
chair() {};
void info() { m->info(); cout<<" Chair"<<endl;}

};
```

```
int main() {

table *wood_table = new table();

wood_table->setMaterial(new wood());

chair *steel_chair = new chair();

steel_chair->setMaterial(new steel());

wood_table->info();

steel_chair->info();

        delete wood_table;
        delete steel_chair;
}
```

output is

```
Wood  Table
Steel  Chair
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31494857_3.html

>

uppose that you have a set of nodes with no null pointers (each node points to itself or to some other node in the set), given a pointer to a node, how to find the number of different nodes that it ultimately researches by following links from that node, without modifying any nodes. DO NOT use more than a constant amount of extra memory spa

Pasted from <http://www.mitbbs.com/article/JobHunting/31495985_3.html

>

1. 给定一个首尾相连的排过序的单链表，首节点最大尾节点最小，给出链表中任意一个节点，要求返回链表中间节点；

2. 一摞未排序的扑克中间有重复，用最有效的方法找出并删除重复者

Pasted from <http://www.mitbbs.com/article/JobHunting/31496467_3.html

>

昨天去某公司面试 Software Engineer 碰到的最后一道题：

有一种新语言，只能做三种操作。
X=0; 给变量赋值为 0；
X++; 递增
LOOP（x）{。。} 给定一个变量值就循环 X 次，循环 block 可以嵌套定义的三种操作。
题目是给定 B，求 A=B-1。

想了很久还是没有想出来。。大家可以帮忙看看有什么思路吗？

Pasted from <http://www.mitbbs.com/article/JobHunting/31496897_3.html

>

发信人: PseudoP (Opium), 信区: JobHunting
标 题: Bloomberg 面经
发信站: BBS 未名空间站 (Wed Jan 20 16:13:13 2010, 美东)

刚收到拒信。financial software developer
面试总的感觉还是不错。interviewers 都还专业，nice。还是很感谢有这个面试机会。
虽然没中，应该是我个人的问题。
background: physics phd
==
1. online skill assessment, Dec 2009
Some like GRE questions.

2. phone interview with 2 people in R&D, Dec 2009
1) how to find 1 missing number from 0 to N in an array of N numbers.
2) brainteaser, 5 jar problems.
3) how to calculate sqrt(N) without using sqrt function. Binary search tree
problem.
4) some behavioral problem. Like, How do you know about BB? Why you wanna
work in BB? Why you wanna work in industry?

3. onsite interview, Jan 2010
1st meet 2 people in R&D
1) train, tunnel, people escaping problem
2) 6 digits number, each changes from 0 to 9. Find the odds that sum of
first
three is the same as the sum of last three. A: 2 do loop.
3) Find 1 missing number from 0 to N. But notice that it is possible the sum
would overflow. Think about a way to avoid the overflow.
4) Tricky problem. I do not think anyone else would know the answer except
the one who gives the problem. Nothing to do with math, statistics.
5) Same 5 jars problem. That is their favorite.

2nd meet a lady in HR.
Ask 15-20 Behavioral problems. Cover most commonly behavioral problems.

3rd meet a senior manager in R&D
Talk a lot about my research.
Ask one question, how to find the first unique number in an array of byte.
and write a code to realize it.

==
希望对后来的有帮助，祝大家好运。

--

※ 修改:•PseudoP 於 Jan 20 16:17:57 2010 修改本文•[FROM: 128.174
.]
※ 来源:•WWW 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM:
128.174.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31497519_3.html

>

发信人: bigdog (大狗 1 号), 信区: JobHunting
标 题: 一道 C 面试题
发信站: BBS 未名空间站 (Sat Jan 23 00:54:52 2010, 美东)

Suppose there is a C function to count and return thhe number of nodes in a linked list.

What cases would you cover in unit tests of this function?

I can only think of two testing cases
(1): An empty list.
(2): An extrem long list with the length of the maximum value of unsigned int.

Any other testing cases?

--

我虽然至今未婚，但日常工作之余一直坚持研究学习关于婚姻、男女、两性的各种心理知识

※ 来源:•BBS 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM:
202.156.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31499799_3.html

>

发信人: timeforce (timeforce), 信区: JobHunting
标 题: bloomberg 第一轮面试
发信站: BBS 未名空间站 (Sat Jan 23 13:28:22 2010, 美东)

早上刚面完，我是 ee 的，所以主要面了些算法和智力题。
1.behavior question, Why you want to join BB?
2.一个 windows 系统，一个 unix 系统，unix 系统里有 100 个数据库，总共 1TB，如何在 1 小时内从 unix 系统转移到 windows 系统中
3.找出一个字符串中最早出现的非重复字母
4.两个鸡蛋测试那层楼丢下来会碎
5.问了些做过的 project 的具体内容

另，我想再联系下我的面试官，我知道名字，怎么找到他的邮箱地址？谢谢。
--

※ 来源:•WWW 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM:
68.36.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31499929_3.html

>

0 0
[上篇] [下篇] [同主题上篇] [同主题下篇]

发信人: jetchen (飞机), 信区: JobHunting
标 题: 问一个链表方面的算法问题 (转载)
发信站: BBS 未名空间站 (Sun Jan 24 00:09:57 2010, 美东)

【 以下文字转载自 CS 讨论区 】
发信人: jetchen (飞机), 信区: CS
标 题: 问一个链表方面的算法问题
发信站: BBS 未名空间站 (Sun Jan 24 00:08:52 2010, 美东)

小弟 EE 的，不是 cs 科班出生，不知道这个问题描述的准不准确：

有一个循环链表 a->d->b->c->e->....->a
每一个节点都是一个整数，且不重复(除了首尾节点外)。

现在这个链表被拆断开了，每 2 个相邻节点被存在一个 cell 里面, 但这些 cell 不是有序的。 就是说链表被拆成了 a->d, c->e,...,d->b,...,b->c,....
我想重新把链表建立起来，应该用什么样的算法?

谢谢.
--

Pasted from <http://www.mitbbs.com/article/JobHunting/31500287_3.html

>

合并两个 BST 要求 O(n+m)时间，n 和 m 为两棵树的大小。有什么好的解法么？

Pasted from <http://www.mitbbs.com/article/JobHunting/31500627_3.html

>

面试了一个小时左右。

1. 用两个 stacks 来实现一个 queue，题不是很难，但是要求逐行念代码，精确到冒号分号，尖括号怎么说
不知道。。。耗了好久。
2. 一些关于 multi-threading，critical section，等等。
3. SQL 的一些问题，我不怎么会，就 skip 了。
4. OOD 问题，如何设计 parking garage，大家有什么好的想法吗？

攒人品中……

Pasted from <http://www.mitbbs.com/article/JobHunting/31501235_3.html

>

给你一个字典 array of strings (you may preprocess it if necessary)

任意一个单词，求最小的 edit distance

一个单位的 distance 定义为:
a. replace a letter
b. delete a letter
c. insert a letter (also at any position)

快速的 code 出来～ 你就可以拿 facebook 面试了

Pasted from <http://www.mitbbs.com/article/JobHunting/31501445_3.html

>

November 02
总结下最近几天看到的一些很有趣的题目

题目 1. LIS. 一个任意的数组，找出一个严格单调递增的最长子序列。
例如: {3,0,1,7,2,4,5,9} –> output： {0, 1, 2, 4, 5, 9}
很简洁巧妙的算法，能在 O(N log N)时间和 O(N)空间做出来！方法就是始终保持一个单增的序列，然后新来的数如果比当前最大还大就 append 在后面，否则在单增序列里面做 binary search，替换相应位置的数。

题目 2. 玻璃杯/鸡蛋 drop 问题。有 N 层楼，假定是在 i 层楼扔鸡蛋，如果没有碎，那么在所有<=i 楼层扔鸡蛋都保证不会碎，反之如果碎了，那么保证在所有 >=i 楼层扔鸡蛋都必碎。通过若干次尝试扔鸡蛋，找到某个鸡蛋碎/不碎的”临界”层。允许你扔鸡蛋的总次数是 D，允许你打碎的鸡蛋数是 B。

问题的描述是：对一组给定的数（N D B），如果存在一个策略保证能在 D B 的限制下，在 N 层楼中找到“临界”层，那么称此(N D B)是 Solvable 的。接下来相关联的三个问题就是:
（a）给定 D，B，求满足（N，D，B）Solvable 的最大的 N_max. 例：D=4，B=1， 策略是从第一层开始一层层往上. N_max=D=4.
（b）给定 F，B，求最小的 D_min
（c）给定 F，D，求最小的 B_min
这个问题相当容易找到看似最优的解，但是绝大部分的方法都不是最优的(最快最高效)。而且最迷惑人的是，(a)(b)(c)三个问题中，必须先从其中某一个下手开始解决，如果你不幸的先从另外的两个问题下手，多半离最优解遥遥无望。

如果你找到了正确的入手点，有了正确的思路，最后的答案会异常的简单！
入手点就是首先解决(a)问题，并且可以递归的来解决：假设 D,B 对应的答案是 F(D,B)，那么考虑在某一层摔一个鸡蛋后，如果碎了，D--,B--，如果没碎就只是 D--，B 不变。
这样很容易写出递归方程，算出 F 关于 D,B 的 table。

题目 3. 经典的概率悖论。3 扇门，一扇背后有羊，你选中一扇门后，现在另外一扇门开了，里面是空的。问你是否应该重新选择。

分析：据观察，有一部分的人坚持认为一定要重新选择，另一部分的人认为是否重新选择都一样。另外少部分的人能看出，这个问题很巧妙的隐含了意识(主观 intention)，信息和概率的关系！

题目 4. 很简单的，N 个数的数组，找出最大的和第二大的数，只用 N+logN-2 的比较次数，不需要额外空间。这个是典型的问题本身就是答案提示的题目--基于比较又有 LogN，很显然思路涉及二分法，继续下去，剩下的问题就仅仅是找一个符合要求的 Implementation 了。

题目 5. 找 N！最后一个非零的数字。巧妙的方法可以在 LogN 时间内找出来，一个 hint 是利用 5^k（和 log_5）划分问题

题目 6. 任务分配，假设有 N 个任务，每个任务需要 W_i 工作量，M 个人，每人每天能做工作量 w_i，如何安排工作，使得所有工作能最快完成。这个问题其实更像一个开放性问题，因为一个合理的贪心策略，最后的结果跟最优结是很接近的(大致上，最多只差一天)。

题目 7. 计算 Fibonacci 数 F(n)，O(n)的算法是很 trival 的。但是有很漂亮简洁的 Log(N)算法，思路是利用 2*2 矩阵表示 Fibonacci 递推式，然后用二分法的思想球矩阵的 N 次方。

题目 8. 一颗 BinaryTree，每个节点有个 NULL 指针，要求把每个节点和在 BFS 中它的下一个节点串起来。其他 BinaryTree 的常见题有比如非递归的实现遍历，用.parent or stack。思考这些题的经验是，对于这一类的树的题目，有很强的递归性/规律性，通常都是 O(N)的复杂度，那么把 N steps 的问题，放在某个单 step 来研究，会把思路变得更清晰。另外一点就是，完全可以假设在做这一单步之前，在做这一步之前的问题已经最大可能的正确解决了，这样能够以一种数学归纳法的思想去利用之前的结论。比如这个题里面，假设节点 i 之前的节点都已经串好了，如何把 i 串到下一个节点。这个问题就是看一眼草图就能知道的了。最后一点经验是，在效率相当的算法的基础上，不同版

本的实现，已经有能够互相启发的地方。
--

Pasted from <http://www.mitbbs.com/article/JobHunting/31502251_3.html

>

发信人: geniusxsy (小尾羊), 信区: JobHunting
标 题: 转一些我 blog 上以前总结题目的日记(三)
发信站: BBS 未名空间站 (Tue Jan 26 15:10:35 2010, 美东)

November 06
一日 3 题

第一题。给一个数组 a[1]到 a[n] : 例如 1，2，3，4，5，6
现在随机生成 a 的一个 permutation: b[1]到 b[n] （例如：3 1 5 2 4 6）
问, a 和 b 数组在每一位上都不相同的概率是多少？假设 a 本身没有重复的数

我的解法：
主问题：F(n) = 给定长度为 n 的 a 数组，b 数组有多少种取法
辅助问题：结果用 f(n)表示。 b 数组是{1….i-1,x,i+1…n}的一个排列，其中 x!=i，满足 a，b 在每一位上都不相同，有多少种 b？例如，a = 1,2,3,4; b 是{1,2,5,4}的一个排列。换句话说，组成 b 的元素中，有且只有一个数不在 a 中。

这样定义了 F(n),f(n)后，很显然有递推关系：
F(n) = (n-1) * f(n-1)   //解释：第一位有 n-1 种选择，任意一种选择后，问题变为一个 n-1 规模的辅助问题

f(n) = F(n-1) + (n-1)*f(n-1) //情况一，在 b 数组的第 i 位置填入 x，考虑剩下的 n-1 个位置，即是一个 n-1 规模的主问题；情况二，i 位置填入非 x 的数，考虑剩下的 n-1 个位置，即是一个 n-1 规模的辅助问题。

简化一下表达式就是:
F(n) = (n-1)(F(n-1)+F(n-2))

第二题，一个 binary tree，逆序打印 BFS 序列。不能同时用两段存储空间（不同时用 queue 和 stack）
解法，用一个 vector（array)模拟 queue+stack。queue 的 push 操作即 vector 的 push_back，等效于 q.pop()+stack.push()的操作则是，vector 的 index 往前走一步！最后把 vector 从尾到头打印一遍即可。

第三题，网上看的答案，超级巧妙，生成一个 0-255 二进制数有多少位是 1 的查询表

```
static int BitSetCount256[256] =
{
#define B2(n) n, n+1, n+1, n+2,
#define B4(n) B2(n), B2(n+1), B2(n+1), B2(n+2),
#define B6(n) B4(n), B4(n+1), B4(n+1), B4(n+2),
B6(0), B6(1), B6(1), B6(2)
}
```

不得不说，这个宏递归的方法用的太妙了！！！
附带赞一个巧妙度略低一些的计算二进制数有多少位 1 的方法

```
int bitSetCount(unsigned int i){
int c=0;
while(i){
c++;
i &= (i-1);  //这一步很赞，每次保证清除最低一位 1；
}
return c;
}
```

--

※ 来源:•WWW 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM:
169.234.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31502237_3.html

>

November 07
学习了 backtrack(回溯法)

之前做了一些回溯的题，比如打印 permutation，打印任意 n 对括号等等，都是瞎蒙的。

还真凑巧，上午做了打印 n 括号的题，下午就看见有人说到回溯法，想想自己还没系统学过这个，找了本基础的中文算法书来看了看，虽然书上讲的很浅显，发现自己貌似瞎蒙还蒙对了思路，呵呵。正好凑巧的是，刚刚看了一点点，网上就有个人问怎么做 Vertex Cover 的问题，正好让我来做做练习。

1. 打印任意合法的 n 对括号：

```
void printParenthes(int N, int left, int right, stack<char> &stk){
   if(left == N && right == N){
      printStack(stk);
      return;
   }
   if(left>right){
      stk.push(')');
      printParenthes(N, left,right+1, stk);
      stk.pop();
   }
   if(left < N){
      stk.push('(');
      printParenthes(N, left+1, right, stk);
      stk.pop();
   }
}
```

2. Vertex Cover（NPC 问题），图 G 中找一个顶点的最小子集，覆盖图的所有边。

```
int current_k = N; //global

void VC(int k, int start_v){
   if(all_edge_covered(G) && k<current_k){
      current_k = k;
      return;
```

```
  }
  if(k == current_k - 1) return; //剪枝
  for(; start_v<=N; start_v++){
    if(!edge_list[start_v].empty()){ //剪枝
      list<int> temp_edge_list = edge_list[start_v];
      clear_edge(start_v,G);
      VC(k+1, start_v+1);
      if(curent_k == k+1) return; //剪枝
      reset_edge(start_v,temp_edge_list,G);
    }//endif
  }//endfor
}//endVC
```

想了想，其中的 for 循环其实是不必的，对于解空间树是子集树的问题，只需要考虑《当前顶点“选”“不选”》两个情况

改进后的算法是：

```
void VC2(int k, int start_v){
  if(k<current_k && all_edge_covered(G)){
    current_k = k;
    return;
  }

  if(k >= current_k - 1) return; //剪枝
  if(start_v == N) return;   //没有下一个顶点了

  if(!edge_list[start_v].empty()){ //如果
    list<int> temp_edge_list = edge_list[start_v];
    clear_edge(start_v,G);
    VC2(k+1, start_v+1);
    if(curent_k == k+1) return; //剪枝
    reset_edge(start_v,temp_edge_list,G);
  }//endif
  VC2(k, start_v+1);   //不选 start_v 这个顶点
}//endVC
```

Pasted from <http://www.mitbbs.com/article/JobHunting/31502231_3.html

>

Dec 05
最近又看到了几道很好的题：

1. 我们知道，从一个数组里找一段(连续的)子数组求最大和，是一道经典的面试题，方法很简单，只要 O(n)的时间。把这个问题变一下，假设是一个循环数组呢？找一个 size<=n 的子数组 with 最大和。

分析，很容易想到第一步，找个地方把循环数组切断，回到了原来的问题，然后在考虑一下额外的情况。额外的情况就是：有可能最大和的子数组是跨越了切断点的？这种情况的最大和怎么求呢？一个 naive 的方法能做到 O(n)，但是需要 O(n)的空间。巧妙的解法就是，注意到所有数的和是固定的，考虑切断后的非循环数组，找一段从首开始+一段从尾开始的两个子数组 with 最大和，等价于找一段子数组 with min sum.

总结，要擅长利用等价性转换问题，从而将新的问题转变为一个已知有好 solution 的旧问题。利用已知的经典问题来解决新问题，可以说是面试题目中相当重要的一个技巧

2. largest rectangular problem：问题是这样的，一个 N×M 的棋盘，上面的数字要么是 1，要么是 0，那么要：a）最大的一个正方形全是 1 填充，b）最大的全是 1 的矩形。

a）是用动态规划做，虽然方法也很好，但是这里就不提了。b）问题感觉上要比 a 难很多，为什么呢，因为 rectangular 比 square 有更大的自由度。不好用 DP 来做，分治也不合适。

这题的奥妙就在于，利用经典问题。什么经典问题呢？其实是另外一道面试题，其本身也是有一定难度的题，题目是：给你一个统计直方图，假设每根柱子都是单位宽度，从图的最左边一个紧挨一个排到图的最右边，求在这个图里找到一个最大矩形，它不跟任何直方柱相交(边缘接触是允许的)。为什么提起这个题呢，故事是这样的，我之前没有做出 O(N*M)解法的 largest rect 题，后来有一天遇到了这个直方图的题目，找到了很漂亮的 O(N)解法，猛然回顾起那道 largest rect 的题，这次就很轻松的搞定了。

3（鸣谢 mittbbs jobhunting 版上的一位面试官贡献自己出的题）有 n 个房间，小偷每天偷一间，偷的规律简单说就是随机行走，如果今天偷了第 i 间屋子，明天有一半的几率偷 i-1，一半的几率偷 i+1，注意如果刚好偷到了边界上，那么第二天只有唯一的选择。

如果你是警察，你只能每天选择一个房间蹲守，并且贼的手段相当高明，偷了一个房间后，没有任何人能发觉该房间是否曾经被偷过。

提示：奇偶性。总结：注意观察题目中隐含的性质。

4. wild card 匹配+搜索：假设你有一个 dictionary(原题中是 URL 集合)，你要搜到到所有与 *a*bc*d 这样的输入所匹配的 words。这里，*是通配符，可以当成是任意个任意字符(包括空)，怎么 预处理+搜索？如果输入是 ???a???b??cde 这类呢？ '?'代表单个任意字符。如果输入是? *的混合呢？

Pasted from <http://www.mitbbs.com/article/JobHunting/31502229_3.html

>

有 m 个 nuts, n 个 bolts，规格大小都不相同
只能 nut 和 bolt 之间比较
怎么把他们排序？要求复杂度最小

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31502045.html

>

居然把时间看错了，原来人家说的是东部时间的 2pm。。。搞得来 11 点打来电话的时候我还紧张了一阵。。。口语说的很烂，脑子也不太灵光。。。

题目其实都不难，behavior 问了你最喜欢的 CS 的东西是什么，我就说算法，然后他还居然提了下我简历上量子计算，估计他不会怎么感兴趣，我就只简单提了一点。然后 coding 题都是很基本的，一个串 in place 删除某些字符，code 完了后，反过来，一个串在某些地方插入字符，期间我脑子短路了一阵子，还好过了一些时间后接上来了。。。然后是个超级老题，数组里面找唯一一个出现了奇数次的整数，我这次很诚实，直接说

我知道这类 trick 的。。。不过他还是让我接着说了一下笨办法怎么做。接下来又是一个老题，楼层扔鸡蛋问题，这个我前面的日记里面都贴过的，所以除了表述上可能有些不清楚外，算法本身肯定是 optimal 的了。接下来问了面试官几个 cliche 的问题就结束了，至少表面上面试官还是比较满意的。。。

整个面了下来居然没有问 design 的题目，也不知道是幸运还是不幸(一次都还没被问过，缺少实战经验)。。。

最大的总结：

看清楚面试时间 + 多练口语表达思路 + 看清楚面试时间。。。

Pasted from <http://www.mitbbs.com/article/JobHunting/31502227_3.html

>

面的一般，关键感觉那面试官比较冷，问他也不说对错，让人挺郁闷。
1. 如何寻找二叉树(not binary search tree)的 least common ancestor.
2. 如何测试一个计算器。
3. 如果你想打电话面试一个人，拿到那个人的简历，却发现他没有提供电话号码，你能想到几种方法找到他的电话号码。

最后一个问题还是挺好的，不知道大家能想到几种方法。 第一个问题我知道如何做，就是找到从 root 到两个节点的 path, 然后比较 path 就可以了。但他问的很细，可能我讲的不是很清楚. 我现在想像这样具体到代码的问题，咋们能不能说在电脑上写好了发到他邮箱，然后再解释啊？ 不然像这个问题，一行一行解释真地挺难的。

Pasted from <http://www.mitbbs.com/article/JobHunting/31502699_3.html

>

very easy, but I think I have to say goodbye

1. deadlock's four condition
2. what's virtual memory
3. how to increase the virtual memory
4. when should we maintain v2p page table, when should we maintain p2v page table.
5. what's mmap & lazyloading. when should we use mmap to allocate memory
6. multi-level page table
7. implement a stack with lock to ensure thread safe

Pasted from <http://www.mitbbs.com/article/JobHunting/31356292_3.html

>

大部分的题都是板上贴过的，再贴贴吧：）
1. C++和 C#的最大区别？
2. 问到了 C++和 Java 在 Runtime Environment（没太听清，可能是这个词）的区别。后来我问问了，好像是在 virtual function 上的区别。我答得是 Java 里所有函数都是 virtual 函数，C++要显示标明。
3. TCP 和 UDP 的区别
前面 3 个应该是根据我简历来问的，其实我 Java 和 C#也没用太多:(

4. 给定一个 int 数组和一个 int 变量叫 sum，返回是否有数组中的 2 个数的和等于 sum。
   这个问题我最开始没处理好正好数组里有一个数等于 sum/2 的情况:(
5. n 级台阶，一次可以上 1 级或 2 级，有多少种上发
   老题目了，fibonacci number
6. reverse the words in a sentence, but don't reverse the words.

Pasted from <http://www.mitbbs.com/article/JobHunting/31344095_3.html

>

. Given a random generator which can generate integer number from 1 to 5 with uniform probability. how to generate number from 1 to 7 with uniform probability.

2. Find the shortest path to convert one string to another using the minimum edits with each transformation string being a valid dictionary word in a dictionary.

for example: for->fork->ford->word->sword

Pasted from <http://www.mitbbs.com/article/JobHunting/31429703_3.html

>

how do you dynamically allocate space for a two dimension array in consecuti
ve bytes?

should be easy

Pasted from <http://www.mitbbs.com/article/JobHunting/31432089_3.html

>

iven N points in a place with their (x,y) co-ordinates. Find two points
with least distance between them.

Pasted from <http://www.mitbbs.com/article/JobHunting/31437667_3.html

>

plz implement a non-recursive post order tree traversal.

I think this is difficult. It is kinda simple for pre-order and in-order, bu
t post-order is tough.

Pasted from <http://www.mitbbs.com/article/JobHunting/31455707_3.html

>

How do you measure context switch time in OS?
any ideas?

Pasted from <http://www.mitbbs.com/article/JobHunting/31465291_3.html

>

ou have a data structure of integers, which can be negative, zero, or positive, and you need to support an API with two public methods, insert(int) and getmedian(). Describe a data structure you would use to support this API and describe the running time of the two methods.

Pasted from <http://www.mitbbs.com/article/JobHunting/31472621_3.html

>

How to sort an array with only {0, 1, 2} possible values in O(n) without
extra space?

Ex: an array {0, 1, 2, 2, 1, 0}

Pasted from <http://www.mitbbs.com/article/JobHunting/31472623_3.html

>

given a string, how to do a string rotation without using extra memory?

Pasted from <http://www.mitbbs.com/article/JobHunting/31473311_3.html

>

Given a list of characters and an int which is the distance between the
same characters

Eg: input- aaaaabbbbcc and distance as 2. One of the outputs can be-
aabaababbcc

Come up with an algorithm and Code it.

好吧，我承认，给我的人说他也不记得了，当时就是挺 confused 的。

来另一个吧：

Q2) Find out if two inputs are Anagrams with HUGE HUGE input (like thousand of terabyte)

Q3) Given lots and lots of points in a 2D space find all the line with most
points on it.

Pasted from <http://www.mitbbs.com/article/JobHunting/31480153_3.html

>

dictionary is given. You have a word which may be misspelled. How will you
check if it is misspelled?

Pasted from <http://www.mitbbs.com/article/JobHunting/31485125_3.html

>

发信人: sumperman (超哥), 信区: JobHunting
标 题: Google interview question
发信站: BBS 未名空间站 (Tue Jan 5 14:59:47 2010, 美东)

Design a system to store heap on multiple machines ? What is avg number of
machines accessed per operation and number of elements stored in a machine ?

First greater number in an array. Given a large array of positive integers,
for an arbitrary integer A, we want to know the first integer in the array
which is greater than or equal A . O(logn) solution required
ex [2, 10,5,6,80]
input : 6 output : 10
input :20 output : 80

Given an N-by-N array of black (1) and white (0) pixels, find the largest contiguous sub-array that consists of entirely black pixels. In the example below there is a 6-by-2 sub-array.

```
1 0 1 1 1 0 0 0
0 0 0 1 0 1 0 0
0 0 1 1 1 0 0 0
0 0 1 1 1 0 1 0
0 0 1 1 1 1 1 1
0 1 0 1 1 1 1 0
0 1 0 1 1 1 1 0
0 0 0 1 1 1 1 0
```

Any ideas?
--

※ 来源:•WWW 未名空间站 海外: mitbbs.com 中国: mitbbs.cn•[FROM: 173.26.]

Pasted from <http://www.mitbbs.com/article/JobHunting/31487235_3.html

>

Given a log file, which contains a series of websites, which the user has visited, find the most frequent path of 3 websites.

e.g: If this is a log file

A B C D E
A C D B E
C D E B A
A C D E B
C D E A B

clearly, C D E in the most frequent website?

Pasted from <http://www.mitbbs.com/article/JobHunting/31493409_3.html

>

. find a pair that add up to a given sum
2. find all phone numbers in the html

pages in a folder (and subfolder).
something else, and self-introduction stuff

Pasted from <http://www.mitbbs.com/article/JobHunting/31493961_3.html

>

2010.1.23 9：00am-11：00am. NYC headquarter
流程：公司介绍+参观公司+面试

我是 CS 的。问的都是 C++编程。一对一。半小时。C++很久之前学过的，恶补了一晚。我跟他说自己
还是 beginner，所以问题不是很难，但是还是不会。。。

1.为什么加入 bloomberg？
2.introduce your experience。
3.比较 c++和 java 的区别。内存管理，garbage class....compiler，编译的作用？（
gcc）(不懂 java。java 貌似编译了以后各个平台上都可以运行吗？virtual machine？C
++编译
了以后的汇编语言只能在特定的系统上运行？）
4.C++中什么 function call delete a object? destructor

5.谈到了 shallow copy deep copy.

```
6.int main()
{
 return main（）;
}
```

will this compile? will this run?

7. 利用一个写好的函数 putchar(char A)which prints out the character you entered. 写一个 putlong (long A).只能调用 putchar，不能调用其他任何函数（可以用 STL，但是不能用 STL 中的函数）。后来还问到了如何 test 你的程序。这一道题折磨了我 15 分钟。

答案在这里：

http://wuhrr.wordpress.com/2007/11/09/how-to-print-a-long-integ
only-putchar/

所有问题不告诉你回答正确与否。他没有固定的几道题要问，你谈到了什么就往深里面问。今天有 50 多人面试，各个背景的都有。CS/EE 的我感觉就问编程，非 CS/EE 的问智力题。再有就
是通知巨突然，周四晚上收到邮件，让周六早上面试。。。

体会：是不是编程牛人，一写程序就知道。。。。这次又去当炮灰了。

Pasted from <http://www.mitbbs.com/article_t/JobHunting/31500097.html

>

Screen clipping taken: 1/28/2010, 5:30 PM

Screen clipping taken: 1/28/2010, 5:31 PM

Screen clipping taken: 1/28/2010, 5:38 PM