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 defragment and compress your memory?

https://www.careercup.com/question?id=3547405

sparse matrix optimization

huffman coding

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.

If it is an array of objects, then each object would store a value and a pointer to the next object.

1. You'll have dead space in the array (entries which aren't currently used for items) taking up memory
2. You'll have to keep track of the free entries - after a few insertions and deletions, these free entries could be anywhere.
3. Using an array will impose an upper limit on the size of the 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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D3376669&sa=D&sntz=1&usg=AFQjCNEXCrGAzBqn6xTgYV73vo3mmzn1fA)

--suffix tree or binary search + Rabin-Karp

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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D3291667&sa=D&sntz=1&usg=AFQjCNG3PtOcH-k86ZNR7-kQdSxZXGpV2g)

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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D3327667&sa=D&sntz=1&usg=AFQjCNFLpIA7GJ-MfylUBgwli0XnnEKlrA)

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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D817692&sa=D&sntz=1&usg=AFQjCNF3bwAtKMUZeNoDqM6a2aHtP95Odg)

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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D382535&sa=D&sntz=1&usg=AFQjCNEQDumcwOsWhuJ69jxoh7mvC3TYNQ)

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](http://www.google.com/url?q=http%3A%2F%2Fdoc.trolltech.com%2Fqq%2Fqq11-mutex.html&sa=D&sntz=1&usg=AFQjCNFSJkoecLmetA5RyfafPSh8RzydsQ)

[QMutex](http://www.google.com/url?q=http%3A%2F%2Fdoc.trolltech.com%2Fqmutex.html&sa=D&sntz=1&usg=AFQjCNGz8bzKHdW9sXA3JsTE_q5Uq0BkIQ) mutex;

void ReaderThread::run()

{

...

mutex.lock();

read\_file();

mutex.unlock();

...

}

void WriterThread::run()

{

...

mutex.lock();

write\_file();

mutex.unlock();

...

}

const int MaxReaders = 32;

[QSemaphore](http://www.google.com/url?q=http%3A%2F%2Fdoc.trolltech.com%2Fqsemaphore.html&sa=D&sntz=1&usg=AFQjCNHAZdXhFRt1GPocvNoaW8bDs-d7lw) 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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D254667&sa=D&sntz=1&usg=AFQjCNG5kXe_u7nLU1Q8Gp-T2-rvgdHc5w)

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](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSuffix_tree&sa=D&sntz=1&usg=AFQjCNEpcLfb_Itp5eXAYLGTEyLpudQ-og)

min/max – heap

Maximum subarray sum problem

**Kadane's 2D algorithm O(N3)**

#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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D209667&sa=D&sntz=1&usg=AFQjCNFq6fPpVzFwWbafxL-MYQ53zG5sgg)

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](http://www.google.com/url?q=http%3A%2F%2Fwww.careercup.com%2Fquestion%3Fid%3D84766&sa=D&sntz=1&usg=AFQjCNGJAOrUkD1P-hcJdzF5Z3-FxLO6pw)

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 performance per operation. Skip Lists allow intermediate nodes in the list to be ``skipped'' during a traversal - resulting in an *expected* performance of per operation

[http://www.csee.umbc.edu/courses/undergraduate/341/fall01/Lectures/SkipLists/skip\_lists/skip\_lists.html](http://www.google.com/url?q=http%3A%2F%2Fwww.csee.umbc.edu%2Fcourses%2Fundergraduate%2F341%2Ffall01%2FLectures%2FSkipLists%2Fskip_lists%2Fskip_lists.html&sa=D&sntz=1&usg=AFQjCNEwIyV1ayz6J8CwbtFR6IT0LmcbpA)

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)

<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