**IMPORTANT**

Merge sort, Stock Max Profit, Max diff where max value is later.

Valid BST,

**get the median from N arrays (can't merge them first).**

confirm open/close parentheses, brackets, etc

How many ways are there to make a dollar out of coins

OOP concepts? What is OOP?

What is assert and how to use it.

What are traits classes, give example in STL

A traits class provides a way of associating information with a compile-time entity.

Traits are class templates to **extract properties** from a generic type.

Examples of traits are the iterator\_traits from the headers <iterator>

Since traits are class templates, they can be specialized. Below an example of the specialization of iterator\_traits for T\*

template<T>

struct iterator\_traits<T\*>

{

using difference\_type = std::ptrdiff\_t;

using value\_type = T;

using pointer = T\*;

using reference = T&;

using iterator\_category = std::random\_access\_iterator\_tag;

};

The expression std::iterator\_traits<T>::value\_type makes it possible to make generic code for full-fledged iterator classes usable even for raw pointers

**Scott meyer has an item dedicated for Traits classes.**

**Item 47: Use traits classes for information about types.**

The traits classes avoid **if-then-else** at run time and does the branching at compile time. For example consider the **advance** stl method, this advances the iterator by the size specified. It’s a generic method and can be used with any iterator types.

template

void advance(IterT& iter, DistT d)

{

doAdvance( iter, d, // call the version of doAdvance that is

typename std::iterator\_traits::iterator\_category()); // appropriate for iter’s iterator

}

// use this impl for random access iterators

template < typename IterT, typename DistT >

void doAdvance(IterT& iter, DistT d, std::random\_access\_iterator\_tag)

{

iter += d;

}

// use this impl for bidirectional\_iterator

template < typename IterT, typename DistT >

void doAdvance(IterT& iter, DistT d, std::bidirectional\_iterator\_tag )

{

if (d >= 0)

{

while (d--)

++iter;

}

else

{

while (d++) --iter;

}

}

template < typename IterT, typename DistT >

void doAdvance(IterT& iter, DistT d, std::input\_iterator\_tag )

{

if (d < 0 )

{

throw std::out\_of\_range("Negative distance");

}

while (d--)

++iter;

}

The forward\_iterator is inherited from input\_iterator, thus this method will work for forward\_iterator also.

The **advance** method calls the appropriate **doAdvance** method, based on the iterator\_category got from the iterator\_traits, the switch occurs during the compile time itself and thus save the clock cycles during run time.

What is priority Queue

What is Dynamic Programming

**What is Trie**

Trie is a wonderful datastructure to store dictionary or keys which are alphabetic strings. The look up time is O(M) where M is the size of the Key being searched.

1. Merge two sorted arrays into second array, both arrays has N elements, but second array size is N \* 2. So you merge both arrays in second array in a sorted way.

**Ans**

**Solution-1**

mergeTwoSortArr(int\* a, int\* b)

Start with the last elements of both the array (f and s) and copy the largest element to the last position of the second array.

Complexity is O(N)

**Solution-2**

Let first array be mPlusN[] and other array be N[]

1) Move m elements of mPlusN[] to end.

2) Start from nth element of mPlusN[] and 0th element of N[] and merge them

into mPlusN[].

1. Find if given text does have matching [ { ( opening and closing. So ({[]}) is valid and ({)} is invalid. They should open and close in proper order.

**ANS:**

While the input char is a opening brace keep pushing into stack. If the char is a closing brace pop element out of stack, compare and check if the input and popped up element make a pair.

1. Rearrange the elements of an array so that all zeroes are at the end of the array, while preserving the ordering of the nonzero elements.

**ANS:**

If (arr[i] == 0)

count++;

else

arr[i-count] = arr[i];

**Solution-2**

Keep coping non zero elements to left and after all the elements are copied replace the last elements with zero.

1. Phone interview, they also used hackerrank.com for a programming question, which asks you to write a function showing the first unique character in a string. pretty straightforward with O(n) complexity

**ANS**

* + Find the count of all the characters in the string.
  + Scan the string from left to right and output the first character having the count as 1.

**Solution-2**: First unique character in a stream

1) Create an empty DLL. Also create two arrays inDLL[] and repeated[] of size 256.

inDLL is an array of pointers to DLL nodes. repeated[] is a boolean array,

repeated[x] is true if x is repeated two or more times, otherwise false.

inDLL[x] contains pointer to a DLL node if character x is present in DLL,

otherwise NULL.

2) Initialize all entries of inDLL[] as NULL and repeated[] as false.

3) To get the first non-repeating character, return character at head of DLL.

4) Following are steps to process a new character 'x' in stream.

a) If repeated[x] is true, ignore this character (x is already repeated two

or more times in the stream)

b) If repeated[x] is false and inDLL[x] is NULL (x is seen first time)

Append x to DLL and store address of new DLL node in inDLL[x].

c) If repeated[x] is false and inDLL[x] is not NULL (x is seen second time)

Get DLL node of x using inDLL[x] and remove the node. Also, mark inDLL[x]

as NULL and repeated[x] as true.

Note that appending a new node to DLL is O(1) operation if we maintain tail pointer. Removing a node from DLL is also O(1). So both operations, addition of new character and finding first non-repeating character take O(1) time.

1. Design a system that keeps track of tickers (name and price) per exchange that: gives the top 20 most recently updated tickers - or something like that.

**ANS:**

1. How to control the behavior of C++ objects and what resides within C++ object
2. I mentioned using a circular buffer as a technique of buffering messages and they asked me to implement one. I had not thought out how to deal with telling the difference between buffer full end buffer empty.
3. char \*a = "hello";  
   char b[] = "hello";  
   char \*c = malloc(12);  
   What do these do, what's the difference in where the memory is stored?
4. int \*p;  
   p=0;  
   p++;  
   printf("%d",p);

char \*a = "hell

1. What's the 'static' keyword in C used for? UNIX: what's a file descriptor, what command do you use to see system operations? What's a thread/deadlock/race condition/etc?
2. What would you do to let a server provide high quality of service to well-behaved clients so that it doesn't get slowed down by a client that can't handle a high rate of traffic?
3. How would you go about designing facebooks suggest a friend feature
4. Reverse a string (in-place), given the input as follows:  
   void ReverseString( char \*str ){  
   /\*meat\*/  
   }

**ANS:**

Simple, swap the elements from both ends.

1. One question was about auto suggestions in a browser  
   How threading works in different flavors of Unix  
   C++ question  
   There were more questions but I can't remember now but I will update this information in the future
2. Find the most populous character in a string.
3. What is the difference between a stack and heap?
4. Give an example of inlined virtual function

**ANS:** Virtual functions can be inlined sometimes. An excerpt from the excellent [C++ faq](http://www.parashift.com/c++-faq-lite/value-vs-ref-semantics.html#faq-31.6):

"The only time an inline virtual call can be inlined is when the compiler knows the "exact class" of the object which is the target of the virtual function call. This can happen only when the compiler has an actual object rather than a pointer or reference to an object. I.e., either with a local object, a global/static object, or a fully contained object inside a composite."

1. What data structure would you use or design to implement a suggestion box for a user searching for someone in a telephone directory
2. find the no. of words in a given string. The words could be separated by space, tab, new line characters.

**ANS**

#include <cctype>

int CountWords(const char\* str)

{

if (str == NULL)

return error\_condition; // let the requirements define this...

bool inSpaces = true;

int numWords = 0;

while (\*str != NULL)

{

if (std::isspace(\*str))

{

inSpaces = true;

}

else if (inSpaces)

{

numWords++;

inSpaces = false;

}

++str;

}

return numWords;

}

1. Second question was to tell the constructor and destructor call order in an inheritance hierarchy
2. Third question was to tell where will the variables will be allocated in memory, there was a global, some local, a global char pointer, a char array, and a local static variable.
3. Fourth question was how to determine the theoretical limit of memory that can be accessed?
4. Dutch National Flag problem. Given a string of R, G, B : RGGBRRBBGG, write a function to arrange it in the order of the characters, all same characters clubbed together, in the order of characters. So the output should be: RRRGGGGBBB
5. const int p = 10;  
   change that value at p by hook or by crook.
6. you are given an int e.g. 10 represented in binary as 1010 flip the bits (2s complement) to 0101.

ANS: if n is the number to be flipped.

return (~n & mask)

This is because the negation will negate all the bits, & with mask will set the non required bit to zero.

How to get mask

Int c = n.

mask=1;

While( c >>=1 )

mask <<= 1;

mask <<= 1;

mask = mask-1;

num = num & mask;

1. How does the compiler interpret const keyword and how it can be fooled.
2. First is a traversal problem, and second was an implementation problem.
3. First was a cache problem, and second was a distributed problem.
4. Rotate array by n elements.

**ANS**

If you want O(n) time and no extra memory usage (since array was specified), use the algorithm from Jon Bentley's book, "Programming Pearls 2nd Edition". It swaps all the elements twice. Not as fast as using linked lists but uses less memory and is conceptually simple.

shiftArray( theArray, M ):

N = len( theArray )

reverseArray( theArray, 0, N-1 )

reverseArray( theArray, 0, M-1 )

reverseArray( theArray, M, N-1 )

reverseArray( anArray, startIndex, endIndex ) reverses the order of elements from startIndex to endIndex, inclusive.

Above is the algorithm to rotate right, below is an example ShiftArray(array, 5, 2)

IP : 1 2 3 4 5

Rev(0,N-1) 5 4 3 2 1

Rev(0,M-1) 4 5 3 2 1

Rev(M,N-1) 4 5 1 2 3

To rotate left use the below algorithm

Rev(0, N-1)

Rev(N-M-1, N-1)

Rev(0, N-M)

NOTE : There is no need to remember the exact steps, we can conclude the exact steps by reversing the array once.

1. Spent a lot of time trying to explain the complexity of building a priority queue is O(n) but the interviewer seemed not buying it.
2. Given a list of integers, all from 1 to n, except for 2 which are marked as "0", find the 2 values that are missing.  
   For example, for n = 5: arr = [1,5,0,0,3] => the two missing values are 2 and 4.
3. buying and selling stock (Leetcode)  
   Fibonacci series. What if integer overflow happens?  
   In linear regression, what is the features are dependent (the feature matrix is not full rank)?
4. The recruiter(engineer) asked me several questions about heap and array.
5. Check if array of int contain two number that sum equal to target number

*Approach 1:*  
The naive way to do this would be to check all combinations (n choose 2). This exhaustive search is O(n2).  
  
*Approach 2:*  
 A better way would be to sort the array. This takes O(n log n)   
Then for each x in array A, use binary search to look for T-x. This will take O(nlogn).  
So, overall search is  O(n log n)  
  
*Approach 3 :*  
The best way would be to insert every element into a hash table (without sorting). This takes O(n) as constant time insertion.  
Then for every x, we can just look up its complement, T-x, which is O(1).  
Overall it takes will be O(n).

*Approach 4:*

What is the array is size is too big and we cannot afford to have additional space and would require inplace algorithm.

Sort the array using inplace sorting algorithm (qsort), it takes nlogn complexity.

Maintain two indexes (leftIndex and rightIndex) to scan the array from both the directions, beginning and end.

While ( left < right )

{

sum = a[left] + a[right]

If (sum == k)

Output the pair and increment the left and right Index.

If (sum < k)

Then we increase the left Index.

If (sum > k)

Then we decrease the right Index.

}

1. How to find the nth element of linked list.
2. Had to think of an efficient algorithm to calculate the average of an array by removing the minimum and maximum
3. You have a given set of prices in an Array, find at which price you should buy and at which price you should sell so that you maximize the gain
4. Find the maximum difference in an unsorted array with the index of max greater than min. array cant be sorted
5. Memory management detail
6. The problem given was that I have an unsorted array. I need to write a simple function which returns the index of the first and second minimum element in the array.
7. The puzzle given was that if I paint a 10\*10\*10 cube from all sides, how many smaller cubes would remain white
8. How would you implement a trie?
9. Merge two sorted arrays, has been mentioned several times in others' report

Let first array be mPlusN[] and other array be N[]

1) Move m elements of mPlusN[] to end.

2) Start from nth element of mPlusN[] and 0th element of N[] and merge them

into mPlusN[].

1. Find the largest difference in an array with the constraint that the larger number has to come later in the array.
2. Verify if you could find two numbers in an array that give the summation equal to a given number N. After I gave out a trivial T(n^2) answer, I was asked if any better method exists.
3. First question was writing a method for detecting if a string is a palindrome. The second question was printing a linked list in reverse. Third was reversing the linked list.

**ANS:** Using recursion the linked list can be printed in reverse.

/\* Function to reverse the linked list \*/

void printReverse(struct node\* head)

{

  // Base case

  if(head == NULL)

    return;

  // print the list after head node

  printReverse(head->next);

  // After everything else is printed, print head

  printf("%d  ", head->data);

}

1. Expain the object oriented concepts.. Polymorphism, Abstraction
2. **Write code to calculate the square root of a number**
3. What is the difference between x++ and ++x, which one is faster, write the functions for these

**ANS:** Prefix is better than postfix.

class X{

private:

int i;

public:

X(){i=0;}

X& operator ++ (){ ++i; return \*this; } //prefix increment

X operator ++ (int unused){ //postfix increment

X ret(\*this);

i++;

return ret;

}

operator int(){ return i; } //int cast

};

1. The questions were: reverse a string in-place in C, what is the difference between x++ and ++x, which one is faster, write the functions for these. Then he gave me some C++ code and asked me: if i define absolutely no method for a class, what does the compiler build by default, then if i have a third-party api call in my constructor and this method throws an exception which i don't catch, what happens + a few other questions i don't remember any more. He ended the interview by writing some C code: there was a string which was hard-coded and he asked me what will different values show (like \*string, or \*(string + 4), string, &string and so on) and what is their type (char / char\* / char\*\* and so on).
2. Finding the biggest value of substring is the most difficult question I faced and my solution is not perfect as well

ANS: The question should be biggest value of sub array, same as the one mentioned below.

1. Find the Largest subarray sum.

<http://www.geeksforgeeks.org/largest-sum-contiguous-subarray/>

int maxSubArraySum(int a[], int size)

{

   int max\_so\_far = a[0], i;

   int curr\_max = a[0];

   for (i = 1; i < size; i++)

   {

        curr\_max = max(a[i], curr\_max+a[i]);

        max\_so\_far = max(max\_so\_far, curr\_max);

   }

   return max\_so\_far;

}

1. Three coding problems in total -- merge sorted array, check if two trees are the same and validate if a tree is a binary search tree. Got one snippet of code in the end and asked me to think of test cases and fix the bug. It mainly completes the task to reverse integer.

Check if two trees are identical

int identicalTrees(struct node\* a, struct node\* b)

{

    /\*1. both empty \*/

    if (a==NULL && b==NULL)

        return 1;

    /\* 2. both non-empty -> compare them \*/

    if (a!=NULL && b!=NULL)

    {

        return

        (

            a->data == b->data &&

            identicalTrees(a->left, b->left) &&

            identicalTrees(a->right, b->right)

        );

    }

    /\* 3. one empty, one not -> false \*/

    return 0;

}

**Validate if a tree is a BST**

|  |  |
| --- | --- |
|  | bool isBSTHelper(BinaryTree \*p, int low, int high) {    if (!p) return true;    if (low < p->data && p->data < high)      return isBSTHelper(p->left, low, p->data) &&             isBSTHelper(p->right, p->data, high);    else      return false;  }    bool isBST(BinaryTree \*root) {    // INT\_MIN and INT\_MAX are defined in C++'s <climits> library    return isBSTHelper(root, INT\_MIN, INT\_MAX);  } |

1. reverse an int (eg. 321 -> 123)
2. She ask about delete the multiple space in an string (eg "abc def gh " -> "abc def gh")
3. Another question about mirror an tree. The answer is simple. Swap each node's left and right child and recursive call its children.
4. Linked list summation(eg. 1->2->3->NULL + 9->8->7->NULL = 1->1->0->4)
5. What is your greatest weakness.
6. If the input is 6, then output should be the series    6 3 10 5 16 8 4 2 1, similarly if input is 10 then it should be 10 5 16 8 4 2 1, and if input is 4 then output is 4 2 1.
7. sorting an arraylist containing 1's and 0's. how to implement LRU cache. How would you implement an auto-complete feature. encoding a word file to save space- any ideas?
8. 1. A sentence has additional spaces between words. Remove the additional spaces, in-place. Like  
   Input - "I am a student", Output - "I am a student".

This problem is like removing zeroes from a string of numbers.

int count = 0;

for (int i = 0; i < SIZE; i++)

{

if (s[i] == ‘ ’)

{

count++;

}

else if(count > 0)

{

s[i-count] = s[i];

// s[i] = ‘ ‘; enable this if space is to be shifted to end

}

}

s[i-count] = ‘\0’;

2. 2 sorted arrays, both has N elements. 2nd array has 2N length, where last N positions are vacant. Merge the arrays in sorted way in the 2nd array.

1. - realization of atoi()  
   - differences between struct and class. what is the default method's, operators and constructor for  
   struct AAA{  
   };  
   - exceptions in contructor and memory leaks.  
   - difference between std::map and std::ordered\_map. Algoritmical difficult...  
   - socket reading. how to read multiple sockets in single thread?
2. 1. why bloomberg  
   2. difference between list and array in general, not specific in a certain language. When when would you like to use array and when would you like to choose list.  
   3. merge two sorted array, one size is N, one is 2N, leetcode has the similar one question  
   4. find a pair of Integers in the array which has max difference. Something like sell stock in the leetcode.
3. Implement a method for a Fibonacci sequence where instead of adding the last 2 elemenT's you add the last n elements. For n=3. 1 1 1 3 5 9...

**ANS:** Check with the interviewer, what should be the initial values of the array.

As the initial values are usually different in different questions it could be 1 1 1 in some cases it could be 1 1 2 etc.

But once the array is initialized with initial values the next step to compute the next elements will always remain the same. The s[i] is the sum of the previous n elements and the s[i-1] is the sum of previous n elements from i-1. Thus to compute s[i] we remove the s[i-n-1] from s[i-1] and then add s[i-1] to it. This formula remains same for any value of n.

If we want to store all the values in a queue then we can use the queue and keep inserting the new values.

This can be stored in a queue

s[i] = s[i-1] – s[i-n-1] + s[i-1];

if we want to just print the values then we can store the latest n+1 values in the queue. Thus the current value will be

nextVal = q.last() – q.front() + q.last();

q.pop();

q.push(nextVal);

1. You're given a binary tree--not necessarily complete or proper--and you need to give each node a "friend" pointer that points to the node to its right in the tree. This node is on the same level but is not necessarily a sibling, which makes the problem a little tricky. The friend pointer of the node farthest to the right on each level should be null.

**ANS: TBD**

1. You're given two arrays of the same size filled with positive integers, and an integer 0 <= N <= 255. You need to determine whether N can be written as the sum of some number from the first array and some number from the second array
2. First I had to implement a doubly linked list, then a method to reverse the order of the nodes
3. Given two arrays of sorted integers, how would you find the minimum difference in O(n) time.
4. Define an algorithm that given a string of words will be able to sort and find most common occurrences of subsequent characters.
5. C++ template metaprogramming (that team was into functional programming and compile time computations). Standard, find nth Fibonacci number using recursive templates.
6. Copy a block of memory from source to destination. You need to consider the overlapping cases. Yet I don't think copy from behind will solve the problem. Because the memory may overlap at the beginning. So a pre-check of overlapping region is necessary. In my view, for a typical copy operation the source should not be overwritten, but the answer allows for that. I was pretty confused..

**ANS:**

This is a problem that happens in practice. There are even hardware instructions that support the copying of memory areas in different directions. If there are source and destination blocks that overlap then you have to compare the address of src and dest and then you will have 3 cases:  
addr(src) == addr(dest): In this case you don't have to do anything  
addr(src) > addr(dest): copy items by progressing from lower offsets to higher offsets  
addr(src) < addr(dest): copy items by progressing from higher offsets to lower offsets  
A real world example to the implementation of this operation is the memmove() C library function.

1. reverse the words in a string
2. Q1 Josephus problem  
   Q2 Find all anagrams of a given word in quickest way possible
3. On the interview day, I only survived 2 rounds.  
   The first round:  
          1. Signal processing questions on my resume. What is LDPC, FFT?  
          2. Return the index of the largest and second largest number of an given array  
          3. Return the occurrence of each letter of a given string in alphabetical order  
          4. If there is a people standing in the middle of a railway and there is a train from a unknown distance coming, there are two stations distanced 300m and 500m in two directions of the railway, he has to decide which direction he should run.  
   The Second Round:  
          1. best time to buy and sell stock  
          2. brain teaser: two robot, same program make them meet each other without knowing their location and no communication between them. Solve it both in 1d and 2d  
          3. Two linked list joint together, find the node they first meet  
              what if there is another list point to one of them, how to find the point which the two lists merge
4. Iterate through std::map and remove items with the specific value.
5. "why bloomberg", "where do you want to be in 5 years"
6. 1. Implement a queue using stacks  
   2. Search through an array of integers and find pairs that add to a target
7. Implement a thread-safe queue push()/pop(). Was presented an offer by the end of the day via my recruiter.
8. Given a web log find the top 10 most visited sequences of 3 URLs.
9. Approximate the square root of a number using binary search
10. Can you give me an example that proves you are good at multi-tasking?
11. How will you implement the following four functions using the hashmap/arraylist:  
    1. Constructor which details how your polynomial is stored  
    2. Adding two polynomials.  
    3. Printing polynomial.  
    4. Finding the derivative of the polynomial.
12. Question1, there are two linked list that might merge, find the merged node.  
    Q2, integer to string  
    Q3. string to integer
13. Can a C++ constructor throw an exception? If so, how should it be handled?   [View Answer](http://www.glassdoor.com/Interview/Can-a-C-constructor-throw-an-exception-If-so-how-should-it-be-handled-QTN_783245.htm)
14. Assume you have a large file with lines of timestamps and IP addresses. Timestamps are ordered, but may repeat and may skip. How do you determine whether there is a time window that has a certain IP address appearing more than k times? How would you solve this if instead you received a stream.
15. 1st Phone interview -  
    - Given a array of numbers and sum k, find a pair in the array that sum to k  
    - Give an algorithm to print the paragraph of text based on the screen size  
      
    2nd Phone interview -  
    - Find the m-th last element in a linked list  
    - Given a large collection of characters(collection will have duplicates as well) and a dictionary, find an efficient algorithm to return the 10 longest words you can form using the characters in the collection. If a character is used in a word, it cannot be used in the next word.  
      
    On site Interview -  
    1st round -  
    Given meetings schedule of the participants. Find a slot when everyone is free  
      
    2nd round -  
    - Each node has up, down, next pointers. Up points to a number less than self, down points to number greater than self. All the nodes are sorted. Flatten the linked list such that it has only next pointer and all the nodes remain sorted.  
    - There is an inflow of ticker symbols and prices. Maintain min, max, last 5 prices seen  
    - Decrypt a string of text. You are given a dictionary and a method that returns a score for a character which is less than 1. If we sum of the scores of all the 26 characters we get 1.  
      
    3rd round - senior manager  
    There are 100 people sitting in a circle. Every second person is opted out. Who wins.  
    4th round -  
    Casual discussion with senior manager  
    5th round -  
    HR talked about experience through out the day and salary and benefits
16. Design an algorithm to find the first unique element in an array.
17. Given a binary tree, how to output it level by level. First just save it in the structure, this would takes space, so just use iterative method to traverse the tree. Using 2 queues, or just calculate the length for 1 level. Next one is ask given a string, and output would delete the spaces, how could u recover the original one. I stuck there not because this is too hard, but I didn’t get the point about what kind of solution he want. Finally he told me that himself. Use encoding method to encode all the original text and this could represent all different characters.
18. 1st round: a white guy and an indian guy. First introduce themselves to me, then talk about myself, then tech problem. Given a binary tree, how to output it level by level. First just save it in the structure, this would takes space, so just use iterative method to traverse the tree. Using 2 queues, or just calculate the length for 1 level. Next one is ask given a string, and output would delete the spaces, how could u recover the original one. I stuck there not because this is too hard, but I didn’t get the point about what kind of solution he want. Finally he told me that himself. Use encoding method to encode all the original text and this could represent all different characters.  
      
    2nd round:  
    Two traditional americans. They both working on a same team and they were just get there. First some behaviors. Then 1st tech is that, given an array of integers, how do you find the first unique element from it. First thought is naive, scan from the first element, then if there is any dup, then move to the next element. Second method is trivial, just try to use a hash map to store all the elements that in the string, if there is dup, just add the count.  
      
    Next question is about how to deal with a random music player. First start with the shuffle algorithm, then talking about using array to deal with the random and non-repeating thing.  
      
    3rd round:  
    a manager from help team. Talking about background, then about some behavior questions.  
      
    Brain teaser: 3 ltr bucket, 7 ltr bucket, how to get 4 ltr water?  
    follow up: 4min hourglass, 7 min hourglass, how to get 9 min?  
      
    4th round:  
    hr interview, just blablabla talking all kinds of things.
19. a function accept string input, and output would delete all the spaces, how could u recover the original one?

**ANS:** **TBD.**

1. Assume there is a graph of API dependencies, a circular dependency would certainly be bad. Create an algorithm that would find any cycles in a graph.  
   Suppose you have checked for circular dependencies and managed to resolve it all, how will you create an order of how the APIs will be linked?

**ANS:** **TBD.**

1. You have a collection of servers that store integers in an unsorted order. Each server can have a different number of integers. If a user wanted to find the median at any given time, how would you find the median?

**ANS: TBD**

1. Explain how the garbage collector works in java and implement it using C/C++
2. Rearrange the elements of an array so that all zeroes are at the end of the array, while preserving the ordering of the nonzero elements.
3. 1. given an array of numbers, find the best time to buy and sell that will give you the largest profit,  
   and that buy has to come before sell.  
     
   I did it first time with 2 for loops and just finding the largest difference, but they want it faster, and I screw up on that part since I didn't remember the things that I could use in programming like maps and tree stuff. anyways, the solution was use a map and just map when to buy to when to sell and another map for when to buy to largest difference. so you can easily know when to buy and sell.  
     
   2. just some basic stuff on C memory structures like whether the passed in value is at stack or heap type of things.
4. Design a data structure with all the features of a stack, but with O(1) lookup for the max element currently in the structure.

**ANS:** Below ans is for minimum and same can be applied for maximum also

Real stack Min stack

5 --> TOP 1

1 1

4 2

6 2

2 2

1. The question is trivial. One is check if a string of parensis is valid or not. The other one is check if a BST is valid.

**ANS: DONE**

1. Do the destructor can throw an exception in c++

**ANS:** NEVER

1. How to remove duplicates from a linked list ? Use of Hashmap  
   **ANS:**

We traverse the link list from head to end. For every newly encountered element, we check whether it is in the hash table: if yes, we remove it; otherwise we put it in the hash table.

Basics of Depth first search and Breadth first search? Which data structure to represent each of them?

**ANS: TBD**

1. What happens when you assign a string literal to a pointer using strcpy

**ANS:** String literal are stored in read only memory and cannot be modified.

1. Implement autocomple. What data structure to be used and why?

**ANS: TBD TRIE**

1. 1) insert a number into an array, 2) find out the missing number in 1 - 1,000,000.
2. From the online application (bloomberg website) to the phone interview took around 4 weeks (a first interview was scheduled within 2 weeks but the interviewer didn't call so we rescheduled). The phone interview was quite straightforward and I was asked the following questions  
   1- Kind of a strange question. **Generate all 3-digit combinations of 1-9 but not repeats in the sense that if you have 123, then 231, 321, etc**. are not allowed. [nested for loops]  
   2- **Reverse a given sentence** (no use of library methods, arbitrary spaces between words). Make it faster. [start at the end and find and print words one-by-one].  
   3- You are given a 3L and a 5L container and unlimited supply of water. How to make 1L?[I found a longer solution but the short on is pour 3L into 5 twice.]  
   4- A program/algorithm which given mL and nL buckets figures out how (if possible) make kL. [we discussed the solution and the interviewer was nice and gave me a hint. Its a DP problem].  
     
   The next day, I received and invitation for on-site interview. The travel schedule part was straightforward and convenient.  
   I was invited to the NY office. There was a group of around 30 people like me. They hold you for a few minuts to issue a security badge. Then there is a tour about all the 31 or so aquariums and fishes at bloomberg. Eventually, there is breakfast and then the interviews come find you. There is 4 rounds of interview. Rounds 1-2 are technical, 3 is management, and 4 is HR.  
   Some questions I remember:  
   1- Find shortest path on a chessboard from low left corner to top right. You can move right and top only and some cells are inaccessible (DFS).  
   2- You have sorted arrays A and B with sizes m and n. Assume at the end of array "A" you have "n" empty slots. How to merge with minimum number of operations (start at the end instead of beginning). Prove it works (induction).  
   3- Some runners running in a field with some "k" sensors. We get events of runnings locations. How to print the top "m" runners? (array of lists or hash table of lists depending on the location of the sensors. each list is one sensor). Some questions on how to identify the ordering within each list.  
   4- CPU Cache implementation. How to implement the least frequently used procedure.  
   5- Scheduling some jobs over "m" servers. Centralized vs distributed implementation? How to implement.  
     
   The management interview was interesting and I liked the two managers. HR was fine too. I did not do any negotiations.
3. The chessboard problem. I first mentioned BFS and use of a Queue but the interviewer kept asking about some data structure with less memory that I could extract due to the simple structure of the graph. I did not understand what he meant. I eventually mentioned DFS and proved it works uses less memory.  
   The cache problem took a long time. I kind of though I failed it at some point. I had no prior knowledge of the topic since I am not a CS guy. I eventually, used an array to store the access time to different items and O(n) search through it to find the least frequently used one. The interviewer did not raise the complexity. He wanted me to write code on a paper (which is hard, esp. in C).
4. Also asked about TCP/IP concepts. Asked about keywords like const,explicit in C++. Inter process communication, pipes, thread and process were few other topics. After the questions, was programming part. 2 questions, 1st was to debug and explain the snippet of the code. 2nd was easy to write code for factorial of an integer.
5. Phone interview:  
       Difference between call by pointer and reference  
       Polymorphism  
       Vptr, virtual table- where is it stored  
       Size of an empty class-why? With a function? With a virtual function?  
       Constructors  
       Copy constructor  
           Static members and static functions  
   In house:  
          Program to reverse an integer array.  
          **Program to generate a random number in the range 1 to 200 with equal probability**. Given a function that always returns 1 or 0.  
         Inheritance  
         Question on logical operators
6. Be prepared with answers to describe your personality, why the company should hire you, and how would you expect yourself to be in 3 to 5 years in the company. Basically everything is quite general.
7. **I cooly answered the graph questions on connecting the levels of a Binary-tree into linked lists.** Then, came a design question of building a ticker feed - I designed using a message passing system fairly easily.

**Method 1 (Extend Level Order Traversal or BFS)**  
Consider the method 2 of [Level Order Traversal](http://www.geeksforgeeks.org/archives/2686). The method 2 can easily be extended to connect nodes of same level. We can augment queue entries to contain level of nodes also which is 0 for root, 1 for root’s children and so on. So a queue node will now contain a pointer to a tree node and an integer level. When we enqueue a node, we make sure that correct level value for node is being set in queue. To set nextRight, for every node N, we dequeue the next node from queue, if the level number of next node is same, we set the nextRight of N as address of the dequeued node, otherwise we set nextRight of N as NULL.

<http://www.geeksforgeeks.org/connect-nodes-at-same-level/>

1. the usual recursive sorting algorithm question : **merge sort, recursive sort**

Parameters: address of array, starting and ending indexes.

IF start\_index == end\_index :

return *// Only 1 record so block is already sorted*

mid\_index = (start\_index + end\_index)/2

*// integer division, no fraction*

*// sort lower half*

sort(array\_address, start\_index, mid\_index)

*// sort upper half*

sort(array\_address, mid\_index+1, end\_index)

*// merge lower and upper half blocks into sorted block*

merge(array\_address,low\_index,mid\_index,high\_index)

### Merge algorithm

Parameters: address of array, starting, mid and ending indexes.

Declare static local array for merging into

WHILE items exist to be merged from both halves:

IF next item in lowerhalf lower than next item in upper half:

copy item from lower half into local array

ELSE:

copy item from upper half into local array

FOR all items not yet copied in either half:

copy item into local array

FOR all items in local array:

copy item back into original block

void merge\_sort(int low,int high)

{

int mid;

if(low<high)

{

mid = low + (high-low)/2; //This avoids overflow when low, high are too large

merge\_sort(low,mid);

merge\_sort(mid+1,high);

merge(low,mid,high);

}

}

void merge(int low,int mid,int high)

{

int h,i,j,b[50],k;

h=low;

i=low;

j=mid+1;

while((h<=mid)&&(j<=high))

{

if(a[h]<=a[j])

{

b[i]=a[h];

h++;

}

else

{

b[i]=a[j];

j++;

}

i++;

}

if(h>mid)

{

for(k=j;k<=high;k++)

{

b[i]=a[k];

i++;

}

}

else

{

for(k=h;k<=mid;k++)

{

b[i]=a[k];

i++;

}

}

for(k=low;k<=high;k++) a[k]=b[k];

}

1. How can you efficiently detect whether two linked lists converged? If they do, how do you identify at which node this occurs?

**ANS:**dd

**Method 3(Using difference of node counts)**  
1) Get count of the nodes in first list, let count be c1.  
2) Get count of the nodes in second list, let count be c2.  
3) Get the difference of counts d = abs(c1 – c2)  
4) Now traverse the bigger list from the first node till d nodes so that from here onwards both the lists have equal no of nodes.  
5) Then we can traverse both the lists in parallel till we come across a common node. (Note that getting a common node is done by comparing the address of the nodes)

1. Given a large collection of characters and a dictionary, find an efficient algorithm to return the 10 longest words you can form using the characters in the collection.

**ANS: TBD**

1. Counting the number of anagrams of one string in another string
2. well all questions are expected, string reverse without reversing words for example " this is sentence" => "sentence is this", write c++ code to do so
3. Write a routine that reverses the word order of an input sentence, ignoring capitalization and punctuation. Whitespace is to be reduced to a single char. The function signature is:  
     
   void reverseSentence(std::string& string)  
     
   Examples  
     
       " the quick brown fox jumps over the lazy dog " becomes "dog lazy the over jumps fox brown quick the".  
       "hello world" becomes "world hello".  
       “ Hi! ” is trimmed to “Hi!”.

**ANS:**

**IP: “Hello how are you”**

1. Reverse the whole sentence

Uoy era woh olleH

1. Then reverse each words

You are how Hello

1. Find if two strings are anagrams of each other

**ANS:**

1. Count the occurrence of each char in first string.
2. Decrement the count of character in the second string, if a chars count is non zero while decrementing then its fine else error.

1. The questions are basic,such as find whether two trees is subtree of each other.

**ANS:**

**Solution:** Traverse the tree T in preorder fashion. For every visited node in the traversal, see if the subtree rooted with this node is identical to S.

bool areIdentical(struct node \* root1, struct node \*root2)

{

    /\* base cases \*/

    if (root1 == NULL && root2 == NULL)

        return true;

    if (root1 == NULL || root2 == NULL)

        return false;

    /\* Check if the data of both roots is same and data of left and right

       subtrees are also same \*/

    return (root1->data == root2->data   &&

            areIdentical(root1->left, root2->left) &&

            areIdentical(root1->right, root2->right) );

}

/\* This function returns true if S is a subtree of T, otherwise false \*/

bool isSubtree(struct node \*T, struct node \*S)

{

    /\* base cases \*/

    if (S == NULL)

        return true;

    if (T == NULL)

        return false;

    /\* Check the tree with root as current node \*/

    if (areIdentical(T, S))

        return true;

    /\* If the tree with root as current node doesn't match then

       try left and right subtrees one by one \*/

    return isSubtree(T->left, S) ||

           isSubtree(T->right, S);

}

1. C++ part: differences between Java and C++ memory model, stack vs heap allocation, difference between pointers and references, friend keyword, private/protected/public visibility modifiers, differences between class and struct.
2. write a function showing the first unique character in a string. pretty straightforward with O(n) complexity..
3. Make a product array , such that each index has product of all the elements in the array except itself   [View Answer](http://www.glassdoor.com/Interview/Make-a-product-array-such-that-each-index-has-product-of-all-the-elements-in-the-array-except-itself-QTN_705075.htm)

**ANS:**

Solution1

Find the product of all the element in the array : allProduct

For (i=0; i < n; i++)

a[i] = allProduct/a[i]

Solution 2

1. Design to display latest socks. Top 10 ticker   [View Answer](http://www.glassdoor.com/Interview/Design-to-display-latest-socks-Top-10-ticker-QTN_705076.htm)
2. Given a pool of strings, any two of them can be joined if last char of first string and first char of second are equal, that joined string will inturn be added to the pool again  
   Eg: abc and cba gives abcba, the common character will be taken once and that applies for for the last and first principle. Multiple characters can occur in the same string.   [Answer Question](http://www.glassdoor.com/Interview/Given-a-pool-of-strings-any-two-of-them-can-be-joined-if-last-char-of-first-string-and-first-char-of-second-are-equal-tha-QTN_705077.htm)
3. Find the Cycle in the graph  
   Difference between tree and graph   [View Answer](http://www.glassdoor.com/Interview/Find-the-Cycle-in-the-graph-Difference-between-tree-and-graph-QTN_705078.htm)
4. write a class for implementing big numbers, really big, which cant be stored in built in types, and all operations involving numbers.' I did write something but it did not look efficient enough.
5. **The second questsion was about implementing Merge Sort in C++**
6. What is an index in database?
7. Binary search tree traversal
8. Write the code for the hash function when the key is a string of 2 characters between a and z.

**ANS:** A wonderful link depicting the method to write hash function for a string.

<http://courses.cs.washington.edu/courses/cse326/00wi/handouts/lecture15/sld013.htm>

for a 2 char string, s1s2 then the hash value should be

int hasher(char\* s)

{

hashValue = s[0] + s[1]\*128;

}

1. They asked me to code a problem that was based on inversions algorithms
2. 1. Two integer arrays, find the numbers that are shared by the two arrays. 2. how to know that a binary tree is a BST?
3. The last question was to find out the largest 2 numbers in a given array.
4. "Find longest palindrome in a given string in less than O(n\*n) time". And I've been struggled for a long time on this.
5. **1. Find out the longest palindrome in a given string in less than O(n\*n) time.**

For example, if the given string is “forgeeksskeegfor”, the output should be “geeksskeeg”.

 We can find the longest palindrome substring in (n^2) time with O(1) extra space. The idea is to generate all even length and odd length palindromes and keep track of the longest palindrome seen so far.

Step to generate odd length palindrome:  
Fix a centre and expand in both directions for longer palindromes.

Step to generate even length palindrome  
Fix two centre ( low and high ) and expand in both directions for longer palindromes.

// This function prints the longest palindrome substring (LPS)

// of str[]. It also returns the length of the longest palindrome

int longestPalSubstr(char \*str)

{

    int maxLength = 1;  // The result (length of LPS)

    int start = 0;

    int len = strlen(str);

    int low, high;

    // One by one consider every character as center point of

    // even and length palindromes

    for (int i = 1; i < len; ++i)

    {

        // Find the longest even length palindrome with center points

        // as i-1 and i.

        low = i - 1;

        high = i;

        while (low >= 0 && high < len && str[low] == str[high])

        {

            if (high - low + 1 > maxLength)

            {

                start = low;

                maxLength = high - low + 1;

            }

            --low;

            ++high;

        }

        // Find the longest odd length palindrome with center

        // point as i

        low = i - 1;

        high = i + 1;

        while (low >= 0 && high < len && str[low] == str[high])

        {

            if (high - low + 1 > maxLength)

            {

                start = low;

                maxLength = high - low + 1;

            }

            --low;

            ++high;

        }

    }

    printf("Longest palindrome substring is: ");

    printSubStr(str, start, start + maxLength - 1);

    return maxLength;

}

2. Design a system that can deliver the newest price of stocks to users

1. When developing financial software, what do you think is different from normal software development?
2. find the character that is \*consecutively\* repeated most times in a string of arbitrary length

**ANS: Simple**

1. Q1, What is a static variable? What is a static function? Think of a case where you can make use of static variable or static function.
2. Design an effective data structure for a phone book, which allows searching by name and also searching by number

**ANS:** I will use two maps.

1. Average interview. Interviewer asked when do we use BST and hash table? What is the complexity of search or insertion for BST?
2. sizeof implementation
3. They asked me how to compress files. And the classical dictionary problem.
4. Given two very very large numbers which cannot hold in 64 bits. How to calculate the final product? Can you optimize your solution?

**ANS: TBD Difficult**

1. How to verify a string made from "{[()]}" is syntactically correct. With each open bracket being closed before any preceding brackets.
2. 1. Find a substring in a string  
   2. A board contains black and white cells. Find the largest size rectangle with edges all white cells.  
   3. Find the subarray in a array of ints with largest sum

**ANS:** Very Interesting I have solved earlier using dynamic programming.

Curmax = max(a[i], curmax+a[i]);

Max\_so\_far = max(curmax, max\_so\_far);

4. Given a process tree. Each process has a pointer to its parent. How can you kill a process? Killing a process means to kill it and all of its descendants

9141262895

39280371

**MAX PROFIT STOCK from array**

#include<climits> // INT\_MIN, INT-MAX

#include<iostream>

#include<vector>

#include<algorithm> // min, max

using namespace std;

int solution(vector<int> &A) {

// write your code in C++14 (g++ 6.2.0)

int gmin = INT\_MAX;

int maxProfit = 0;

int N = A.size();

for (int i = 0; i < N; ++i)

{

// Find the minimum element

gmin = min(gmin, A[i]);

// Find the max difference upto now

maxProfit = max(maxProfit,A[i] - gmin);

}

return maxProfit;

}

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

{

//int arr[] = {2,3,-1,4};

vector<int> v={2,3,-1,4};

cout << "Solution=" << solution(v) << endl;

return 0;

}

**Maximum Sub array**

int maxSubArray(vector<int> &A) {

// write your code in C++14 (g++ 6.2.0)

int maxEndingHere, maxSubArray;

maxEndingHere = maxSubArray = A[0];

int N = A.size();

for (int i = 1; i < N; ++i)

{

// Find the minimum element

maxEndingHere = max(A[i], maxEndingHere+A[i]);

// Find the max difference upto now

maxSubArray = max(maxSubArray,maxEndingHere);

}

return maxSubArray;

}

**Example:**

i/p: [2,3,-9,4]

o/p: 5

i/p: [2,3,-1,4]

o/p: 8

C++ custom comparator multiple fields and parameters

bool CompareData(const T& a, const T& b)

{

if (a.PrimaryCondition < b.PrimaryCondition) return true;

if (b.PrimaryCondition < a.PrimaryCondition) return false;

// a=b for primary condition, go to secondary

if (a.SecondaryCondition < b.SecondaryCondition) return true;

if (b.SecondaryCondition < a.SecondaryCondition) return false;

// ...

return false;

}

Find all combinations of ‘R’ numbers from ‘N’ numbers.

NCR = N! / ((N-R)! R!)

NCR = N-1 C R + N-1 C R-1

NC0 = 0

NCN = 1

**Knapsack Problem**

<https://www.geeksforgeeks.org/knapsack-problem/>

The first question is given binary tree. populate the next node for each tree node.  
the node has {val, left, right, next} properties. similar to Populating Next Right Pointers in Each Node II in leetcode.

Use the Level order traversal – BFS to solve this problem.

**False Sharing – Very important question asked at multiple places.**

<https://en.wikipedia.org/wiki/False_sharing>

struct foo {

int x;

int y;

};

static struct foo f;

*/\* The two following functions are running concurrently: \*/*

int sum\_a(void)

{

int s = 0;

for (int i = 0; i < 1000000; ++i)

s += f.x;

return s;

}

void inc\_b(void)

{

for (int i = 0; i < 1000000; ++i)

++f.y;

}

The above false sharing problem can be solved by using alignas. Assuming the size of the cache line is 64 bytes.

struct foo {

alignas(64) int x;

alignas(64) int y;

};

<https://en.cppreference.com/w/cpp/thread/hardware_destructive_interference_size>

hardware\_destructive\_interference\_size:

1) Minimum offset between two objects to avoid false sharing. Guaranteed to be at least alignof([std::max\_align\_t](http://en.cppreference.com/w/cpp/types/max_align_t))

struct keep\_apart {

alignas(std::hardware\_destructive\_interference\_size) [std::atomic](http://en.cppreference.com/w/cpp/atomic/atomic)<int> cat;

alignas(std::hardware\_destructive\_interference\_size) [std::atomic](http://en.cppreference.com/w/cpp/atomic/atomic)<int> dog;

};

This is portable way of accessing the cache line size.

**Alignas**

<http://en.cppreference.com/w/cpp/language/alignas>

The alignas specifier may be applied to the declaration of a variable or a non-bitfield class data member, or it can be applied to the declaration or definition of a [class/struct/union](http://en.cppreference.com/w/cpp/language/classes) or [enumeration](http://en.cppreference.com/w/cpp/language/enum). The object or the type declared by such a declaration will have its [alignment requirement](http://en.cppreference.com/w/cpp/language/object#Alignment) equal to the strictest (largest) non-zero *expression* of all alignas specifiers used in the declaration.

Alignas(type) allows aligning of variable to be same as that of other type. For example

Struct st\_t{

alignas(double) int x;

}

Struct ct\_t{

alignas(64) int x;

}

Sizeof(st\_t) 🡺 this is same as sizeof(double)

Sizeof(ct\_t) 🡺 this is same as 64 bytes

Whereas the sizeof(st\_t::x) is still 4 bytes

Alignas adds the padding so that the struct/class/enumeration gets aligned to the type specified in Alignas

<https://stackoverflow.com/questions/38844505/does-alignas-affect-the-value-of-sizeof>

// every object of type sse\_t will be aligned to 16-byte boundary

struct alignas(16) sse\_t

{

float sse\_data[4];

};

// the array "cacheline" will be aligned to 128-byte boundary

alignas(128) char cacheline[128];

**Promise**

<http://en.cppreference.com/w/cpp/thread/promise>

The class template std::promise provides a facility to store a value or an exception that is later acquired asynchronously via a [std::future](http://en.cppreference.com/w/cpp/thread/future) object created by the std::promise object.

Each promise is associated with a *shared state*, which contains some state information and a *result* which may be not yet evaluated, evaluated to a value (possibly void) or evaluated to an exception.

This above link has very good example code to illustrate the promise and future at work.

**SIG TEST - 1**

<https://stackoverflow.com/questions/39567821/optimize-solution-to-split-an-array-into-3-parts-at-indices-p-q-such-that-cost-i>

The problem is as follows:

* An array contains N integers.
* Index starts from 0.
* We have to break this array into 3 parts at breakpoints P and Q, where: 0 < P < Q < N-1.
* **P and Q cannot be adjacent**, that is: Q-P > 1.
* The cost of this breaking is arr[P]+arr[Q].

An array can be broken in many ways using various legal combinations of P and Q, each having its own cost. Find the minimal cost.

For example, an array

arr = [5,2,4,6,3,7]

we can break as follows:

P, Q -----> Cost

1, 3 -----> 2 + 6 = 8

1, 4 -----> 2 + 3 = 5

2, 4 -----> 4 + 3 = 7

Hence minimum cost = 5.

**Solution**

int P = 1;

int result = MAX\_VALUE;

for(int Q = 3; Q < n - 1; Q++){//Start the loop with Q = 3

result = min(result, data[Q] + data[P]);

if(data[P] > data[Q - 1]){

//Update min only with the last element to maintain the constraint P - Q > 1

P = Q - 1;

}

}

return result;

Time complexity : O(n), space complexity O(1)