```
my initial questions:
1. group anagrams
2. sum 3 int to 0

NDA files:

1.Marquette 13200 + tuition
2.minesage: RMB90,000  RMB4000
3.Facebook recruiter Raj .. contacted me by email

basics:
1. Write a function to determine if a string is a palindrome
1. printing binary search tree in reverse,
2. finding k-th largest element in O(N) without modifying the node, and then
   same k-th largest element in log(N) time keeping the size of the subtree in each
node.
4. Given an unsorted array, extract the max and min value using the least number of
comparison ()
5. Two texts are considered to "match" if they have a common substring of at least
length n. Describe an algorithm to determine if two strings are matches.
   LIS problem
6. Add two binary numbers represents in strings, those could be large values. (use
number and carry, in careercup)
7. Implement a simple regex parser.  (http://leetcode.com/2011/09/regular-expression-
matching.html)

9. find 3 elements in an array that sum to 0
(http://stackoverflow.com/questions/2070359/finding-three-elements-in-an-array-whose-
sum-is-closest-to-an-given-number)
10. group anagrams (http://stackoverflow.com/questions/396005/algorithm-for-grouping-
anagram-words)
11. Print a tree, level by level. (BFS, one queue, if need print after one level, need
two queues, one queue one level, and swap )
12. Write all solutions for a^3+b^3 = c^3 + d^3, where a, b, c, d lie between [0, 10^5]
13. Maximum profit for buying selling a stock given an array of prices for n days.
14. Isomorphic trees
15. The problem was to convert roman literals into its equivalent numeric value

18. Write a function that takes 2 arguments: a binary tree and an integer n

19. Add up two big integers represented in arrays (or string)
20. Design a system to support Facebook status update
21. Design the recommendation system for search keywords
22. MapReduce: counting the most frequent words from a terabytes file. check how google
does this
23. Dynamic programming
24. Trie
25. Graph
26. Design
27. Linear algebra, polygon convex, complex,

28. Reversing a linked list,
29. Write the actual code to parse a regular expression including "*",
    which stands for 0 or more characters, "+", which stands for 1 or more characters,
and ".",
    which stands for 1 exact character
    (http://www.params.me/2011/11/regex-parser-implementation-in-java.html)
    (http://leetcode.com/2011/09/regular-expression-matching.html)

30. Algorithm question invovles matrix and finding paths. Requires Dynamic Programming
```

31. Finding k-th smallest element in a tree
32. Given N credits cards, determine if more than half of them belong to the same person/owner.
    All you have is an array of the credit card numbers, and an api call like isSamePerson(num1, num2).

33. compress a string
34. Given an integer, return all sequences of numbers that sum to it. (Example: 3 -> (1, 2), (2, 1), (1, 1, 1))
35. Given 2 arrays sorted in ascending orders (a[n] and b[n*2]) with the second half of array a[] being empty,
    create a function that merges both arrays into a single sorted array

36. Given an unsorted array, extract the max and min value using the least number of comparison (n-1 for one, n-1-n/2 for another)
37. Given an unsorted array, extract the second max/min using the least number of comparison (n+logn-2 using tournament, pairwise comparison)
38. Write a palindrome-checking function (two pointers)

39. What is the most exciting thing about the current project Im working on (Behavior questions)

40. Given a Pre-Order and In-Order string of a binary tree, can we and if we can, construct the "Post-Order" String. (Yes, get root from pre, and divide in-order)
41. Longest common substring (use DP) -> http://www.geeksforgeeks.org/dynamic-programming-set-4-longest-common-subsequence/
42. Implement a read/write lock (using one mutex and semaphores class, practice coding [http://doc.qt.digia.com/qq/qq11-mutex.html])
    (In c#, lock() keyword only allow one thread in the critical section; ReaderWriterLock class allow read often, few write, multiple
    read, one write, performance is not good. But in .net 3.5 ReaderWriterLockSlim performance much better)

        another: Implement a read/write lock, given a mutex that has lock() and trylock() interface

43. An operation "swap" means removing an element from the array and appending it at the back of the same array. Find the minimum number of "swaps" needed to sort that array.
    Eg :- 3124
    Output: 2 (3124->1243->1234)
    How to do it less than O(n^2)

44. 1. I, V, X to value function (I think just Convert.Int32('I'-'A') + 65)
    2. string distance, given add, remove and replace operations (use DP)

45. Write all solutions for a^3+b^3 = c^3 + d^3, where a, b, c, d lie between [0, 10^5] (my own solution: it must have a<c<d<b, so first two loops for a, b, and the third loop
    c = a+1, d=b-1 initially, while(c<d) check if good, then we found one. Runtime is O(n^3))
46. Print a list (linked-list?) in reverse (if array, just from n-1 down to 0, if linked-list, use a stack, and then pop out?, or just string, concat together in reverse order)
47. Isomorphic trees (just check shape, solution: http://tech-queries.blogspot.com/2010/04/isomorphic-trees.html)
48. Print all subsets of a set (totally 2^n, cool solution: http://stackoverflow.com/questions/7206442/printing-all-possible-subsets-of-a-list)

49. You are given an array of integers. Find all the combinations of the numbers of the array, that sum to another number(might be different for different combination)
    from the array. One property of the array: The maximum number of the array will not

be much greater than the others.
    (see: http://stackoverflow.com/questions/12837431/find-combinations-sum-of-elements-in-array-whose-sum-equal-to-a-given-number)

50. Max sum of non-adjacent (means element index, not value) element combination in an all positive integer array (use DP, maxsum[i] = max(maxsum[i-2] + a[i], maxsum[i-1]))
    (http://code-em-up.blogspot.com/2012/07/question-given-unsorted-array-of.html)
51. Max sum of non-adjacent (means element index, not value) element combination in an all positive integer array, and max sum no more than K
    {http://stackoverflow.com/questions/10261565/max-sum-in-an-array-with-constraints}
52. Given two sorted arrays, no duplicates, could be any size, return the median number in faster than O(n) time. I think use binary search on two arrays at the same time.

53. Convert roman literals into its equivalent numeric value / convert numeric value to roman literals

54. Write a function that takes 2 arguments: a binary tree and an integer n, it should return the n-th element in the inorder traversal of the binary tree
55. A variation of the classic knapsack problem (DP)
56. Balanced binary search trees, A simple regex parser
57. Sort a binary tree and give the complexity of that, Reverse a linked list

58. Suppose we can translate numbers into characters: 1->a, 2->b, ...26->z given an integer, for example, 11223, output every translation of the number (use recursive)
59. You have a file consists of billions of records. It cannot fit into memory, so you need to reverse every word in that file and save to another file
60. write an iterator function that returns next node in inorder traversal sequence in binary trees. You can write an init() function to initialize what you need
   (I think in init() traversal tree in order, put nodes in linkedlist, just a head node or put nodes in a queue, and then in next() function just call head.next or dequeue if using queue)
61. Given array of elements of 3 types. Sort it (O(n), use counting sort, if already know 3 type, such 0,1,2 then we can use national flag algorithm)
   (http://www.geeksforgeeks.org/sort-an-array-of-0s-1s-and-2s/)
62. Find all subsets of size k from given set
(http://karmaandcoding.blogspot.com/2011/12/find-all-possible-subset-of-array-of.html)

63. Given list of revision numbers and build statuses (OK or FAILED). Example : { (10001,OK) , (10002, OK) , (10003, FAILED) , (10004,FAILED) }
    Find number of revision where build has failed in first time (if once failed, it will fail for any following build, then we can use binary search, if not, we can just search in O(n))

64. Data structure for numbers that supports 2 operations: insert and get_median. (use min-max-heap)

65. Write a function to check if polygon is simple based on given list of points
   (my thoughts: consider polygon points as a string, since we can rotate polygon points once then we still get the same polygon, so the problem becomes:
    Given two string s1 and s2 how will you check if s1 is a rotated version of s2 ? Just concat s1 or s2 on itself then check if s2 or s1 is substring() of concated string)

66. How would you store and search 1 million names (use Trie -- prefix tree, it's storage efficient and have good search time)

67. Given a m*n grid starting from (1, 1). At any point (x, y), you has two choices for the next move: 1) move to (x+y, y); 2) move to (x, y+x);
    From point (1, 1), how to move to (m, n) in least moves? (or there's no such a path)

(solution: like find greatest common divisor, starting from m,n down to 1,1. If m==n but m>1 no path, if m > n, m = m - n, n = n; else m = m, n = n - m; else m==n==1 return true)

68. Write a program to print fibonacci series (using DP)

```
function int fibonacci(int n)
    {
            int[] f = new int[n + 1];
            f[0] = 1;
            f[1] = 1;
            for(int i = 2; i<=n; i++)
            {
                    f[i] = f[i-1] + f[i-2];
            }
            return f[n];

            //or only two vars, don't need a auxiliary array
            int prev = 1; int prevprev = 1;
            for(int i = 2; i<=n; i++)
            {
                    int temp = prev + prevprev;
                    prevprev = prev;
                    prev = temp;
            }
            return prev;
    }
```

69. A period of time where users login and logout, given a sets of login and logout time pairs, write a function that can show the number of users online at any given time

(good thoughts: Sort all the login times in one list, sort all the logout times in another list. This is O(NlogN) preprocessing. Now when a query comes along, do a binary search

on both lists to count the number of logins and the number of logouts prior to the specified time. Then the number of logged in users is logins minus logouts.

This query answering process is O(logN). So for N elements and D queries, we achieve O((N+D) log N) when we count both preprocessing and query cost.)

or check it out about: Interval tree

70. Implement the "see and tell" (or look and say) algorithm with a given seed number x and a number of iterations y. Output the result on iteration y.

(Definition: The integer sequence beginning with a single digit in which the next term is obtained by describing the previous term.)

(check this solution: http://rosettacode.org/wiki/Look-and-say_sequence) like 1, 11, 21, 1211, 111221

71. Function to compute the number of ways to climb a flight of n steps. Taking 1, 2, or 3 steps at a time. Do it in Linear time and constant space. n = 3. 1 1 1 1 2 2 1 3 Ans = 4

(use dp: f(n) = f(n-1) + f(n-2) + f(n-3), in order to get to n, we probably get to n-1, or n-2, or n-3, in each state, think about how we can get to n, it's only one way, that's plus 1 or 2 or 3 to get n,

so totally there are 3 cases, and put them together)

72. Interweave a linked list. Do it in Linear time and constant space. Input: A->B->C->D->E Output: A->E->B->D->C

(definition: interweave means head and tail two pointers, write head and tail and then move next for head pointer, and move back for tail pointer, and then write head and tail until only one or no more elements left)

(solution: go the mid, divide into two halves, reverse the second half, then merge one by one) http://www.youtube.com/watch?v=23DIhKga9MA

73. Get numeric number out of a roman string, linear time Given: mapping I = 1, V = 5,

X = 10, L = 50, C = 100, D = 500, M = 1000 Input/Output: II = 2, III = 3, IV = 4 VI, VII, VIII IX = 9 XL = 40, XLIX = 49

74. Merge 'k' sorted arrays, each array may have max 'n' elements

75. Giving lots of intervals [ai, bi], find a point intersect with the most number of intervals

76. Intersection of n sets without using a hash table (http://www.glassdoor.com/Interview/Intersection-of-n-sets-without-using-a-hash-table-QTN_238475.htm)

77. Implement a LRU(Least Recently Used) cache

78. Implement a function to compute cubic root. what is the time complexity? (use newton method, Xn+1 = 2/3 * Xn + a/3 * Xn^2)

79. Given a file with 3-letter words, print all 3x3 with each row, column and diagonal being one of the words from given file
(some thoughts: http://stackoverflow.com/questions/7621733/arranging-3-letter-words-in-a-2d-matrix-such-that-each-row-column-and-diagonal)

80. Given n+1 buckets with n of them with ball inside and move(a,b) function, that moves ball from bucket a to bucket b. Each ball has a different number from [1,n] on it.
Move balls, so each bucket has a ball with matching number in it
(my thoughts: since have n+1 buckets and only n numbers, it means has one empty bucket. suppose the empty is the last one. We only need scan buckets once. every time just check if buckets[i] == i,
if not, we know we need swap buckets[i] and buckets[buckets[i]], how to swap, we just call move three times. move(buckets[i], n); move(i, buckets[i]); move(n, i), Just normal swap)

81. Find the center of graph(vertex, that is connected with every other vertex, but edges are directed to the center of graph)
(my thoughts: for All E(Edges) in G (graph), each E is a pair of vertices (u, v), so scan each (u, v), add (v, count+1) into a hash table,
then scan hashtable to find out if exist count >= n-1, if has, then that one is the center of graph)

82. Given a matrix with 1's and 0's, find the number of groups of 1's. A group is defined by horiz/vertically adjacent 1's.
(solution: Blood Fill, add a 2d array to record which one has been visited. For any item, if it's 1 and not visited, then groupNumber++, then use BloodFill algorithm to fill the 2d array)
(http://www.glassdoor.com/Interview/Given-a-matrix-with-1-s-and-0-s-find-the-number-of-groups-of-1-s-A-group-is-defined-by-horiz-vertically-adjacent-1-s-QTN_234777.htm)

83. Given a matrix of numbers in which some may be zero. If a cell contains a zero, set all the cells in the corresponding column and row to zero
(only figure out which row has zero and which col has zero (two arrays), and then scan the original array, if that row or col has zero, set that cell 0. only m+n more space, and mn time complexity)

84. given the utitlies getFriend(User u) and areFriends(User u1, User u2), write the function which takes as parameter the array of users and return a bool
saying if you can divide the users in 2 groups s.t. if u1 and u2 both belong to a certain group, they are not friends.
(solution: this tests Bipartite Graph(¶þ·ÖÍ¼), we construct a graph first then check if this graph is a bipartite graph.

the condition is if only if the graph can be two colorable or only if it has not odd length cycles. use BFS, label color
MayBe:
http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/breadthSearch.htm)

85. Implement atof function. eg., +3.5e-2, .03e1, 1e1, 0.0

86. Given an array of integers, now we want to erase all 0's (can be other value), and we want the result array condensed, meaning no empty cell in the array
(solution: if create a new array, it's pretty easy, just scan the org array and add non-zero value into new array. Another way to do is in place. we can have a variable zero_num to store how many zeros so far
and then move current non-zero value back to i-zero_num position and then set 0 to current position)

87. Open-ended systems/design question on storing and searching zillions of status updates

88. Calculate x ^ y in O(log n) -> power(x, y)

89. FInd the maximum sum of a sub-sequence from an positive integer array where any two numbers of sub-sequence are not adjacent to each other in the original sequence. E.g 1 2 3 4 5 6 --> 2 4 6
(use dp: max(n) = Math.max(max(n-1), max(n-2) + a[n])) max(0) = a[0], max(1) = max(a[0], a[1])

90. How can one implement a queue with only a stack implementation?
(use two stack

```
public class Queue<E>
{
        private Stack<E> inbox = new Stack<E>();
        private Stack<E> outbox = new Stack<E>();
        public void queue(E item) {
                inbox.push(item);
        }
        public E dequeue() {
                if (outbox.isEmpty()) {
                        while (!inbox.isEmpty()) {
                           outbox.push(inbox.pop());
                        }
                }
                return outbox.pop();
        }
}
```

91. Given a list of words with a same size and a big string that contains one of the permutation of all the words combined(say p), find the startindex of the string p in the big string
(solution: use hashtable to store <key=word, value=word occurrences> for words. the permutation of all the words has fixed length (n * wordSize), so for big string, we start from index 0 and pick permutation size,
and then check if this picked string is a permutation of org words, how to check, build a hashtable for picked string, and compare hashtable to see for any key they have some value)

92. How would you implement a method to tell whether or not a string matches a regex that consists of lower case letters and *s (Regex parser)
93. Write the actual code to parse a regular expression including "*", which stands for 0 or more characters, "+", which stands for 1 or more characters, and ".", which stands for 1 exact character.

94. Print out all prime numbers in a given string. abc2134kd31 -> 2, 13, 3, 3
    (solution: need a variable to know what's the num so far, and then for this specific num, e.g. 213, we need only check numbers 213, 13, 3, don't need check 2, 1, 21 which have been checked before getting this number)

```
        static bool isPrime(int x)
    {
                for (int i = 2; i <= Math.Sqrt(x); i++)
                {
                        if (x % i == 0)
                                return false;
                }
        }
```

95. Implement division without using multiplication or division. It should work most efficient and fast.

```
        //demominator is ·ÖÄ¸£¬
        int divide(int numerator, int denominator) {
                int mask = 0x1; //1
                int quotient = 0;
                //left shift to find out a value just bigger than numerator
                while (denominator <= numerator) {
                        denominator <<= 1;
                        mask <<= 1;
                }
                //then right shift go back to see if numerator is greater than denominator, if yes, then update numerator and quotient
                while (mask > 1) {
                        denominator >>= 1;
                        mask >>= 1;
                        if (numerator >= denominator) {
                                numerator -= denominator;
                                quotient |= mask;
                        }
                }
                return quotient;
        }
```

96. Implement a suggestion function that generates alternative strings for given password strings like "facebook" => "F@ce?00k" and "f?c€Bo0K" or sth.

97. search needle in haystack problem (search substring problem)

98. Implement a function rotateArray(vector<int> arr, int r) which rotates the array by r places. Eg 1 2 3 4 5 on being rotated by 2 gives 4 5 1 2 3.
    (solution: reverse the whole array first, then reverse the first part, and the second part respectively)

99. Implement a function string balanceParanthesis(string s); which given a string s consisting of some parenthesis returns a string s1 in which parenthesis are balanced and differences between s and s1 are minimum.
    Eg - "(ab(xy)u)2)" -> "(ab(xy)u)2", "))))(((" -> ""

100.How will you design TinyUrl? Keys can be generated in base 36, assuming 26 letters and 10 numbers. or base 62 (26 + 26 + 10)
101.How will you design facebook newsfeed. Focus was on a design which could handle the huge number of status updates and display them on each of the user's friend's wall

102.Find Kth smallest element in a BST.
    (solution: traversal BST in-order until reaching the Kth element)

103.Given a Binary Search Tree, iterate over the elements without using recursion (in-order, use stack)

104.Given a set of non-overlapping integer ranges (1,3) (5,8), etc., and an input integer, what is the best way to organize the data and allow for quick search based on the input, etc.

(sort by the lower number?)

105.Given a telephone number, find all the permutations of the letters assuming 1=abc, 2=def, etc.

```
        static void PrintPhoneNumberPermutations(string phoneNum)
        {
                Dictionary<char, string> m = new Dictionary<char, string>(10);
                m['0'] = "0";
                m['1'] = "1";
                m['2'] = "abc";
                m['3'] = "def";
                m['4'] = "ghi";
                m['5'] = "jkl";
                m['6'] = "mno";
                m['7'] = "pqrs";
                m['8'] = "tuv";
                m['9'] = "wxyz";

                List<string> finalStrs = null;
                foreach (char c in phoneNum)
                {
                        List<string> newStrs = new List<string>();

                        // First time in loop
                        if (finalStrs == null)
                        {
                                finalStrs = new List<string>();
                                foreach (char d in m[c])
                                {
                                        finalStrs.Add(d.ToString());
                                }
                                continue;
                        }

                        // Sebsequent times in loop do permutations
                        foreach (char d in m[c])
                        {
                                foreach (string prevStr in finalStrs)
                                {
                                        newStrs.Add(prevStr + d);
                                }
                        }

                        finalStrs = newStrs;
                }

                foreach (string s in finalStrs)
                {
                        Console.WriteLine(s);
                }
        }
```

106.Print a singly-linked list backwards, in constant space and linear time.

(solution: reverse, print, reverse back)

```
        public class Node
        {
```

```
                public Node Next;
                public int Value;
        }

        public void PrintMain(Node head)
        {
                Node newhead = Reverse(head);
                Print(newhead);
                head = Reverse(newhead);
        }

        public Node Reverse(Node head)
        {
                Node newhead = null;
                while(head != null)
                {
                        Node tempNode = head;
                        head = head.Next;
                        tempNode.Next = newhead;
                        newhead = tempNode;
                }
                return newhead;
        }

        public void Print(Node head)
        {
                while(head != null)
                {
                        Console.Write(head.Value)
                        head = head.Next;
                }
        }
```

107.Convert a binary search tree to a sorted, circular, doubly-linked list, in place (using the tree nodes as the new list nodes).
    (http://cslibrary.stanford.edu/109/TreeListRecursion.html)
        (solution: use recursion, the basic idea is to convert left tree and right tree and root node into 3 doubly-linked lists, and then connect these 3 lists.)

```
        public class Node  //this class is used as tree nodes and linkedListNode, in linked list, Left means previous node, Right means Next node
        {
                public Node Left;
                public Node Right;
                public int Value;
        }

        //return the head node for doubly-linked list, but the parameter is the root of tree
        public Node TreeToList(Node root)
        {
                if (root == null) return null;

                //try to get linkedlist from left tree and right tree
                Node lefthead = TreeToList(root.Left);
                Node righthead = TreeToList(root.Right);

                //after convert these two subtrees into list, we need make the root of tree into a doubly-linked list, the 'Prev' and 'Next' points to itself
                root.Left = root;
                root.right = root;
```

```
                //now we have 3 doubly-linked list, we need connect them together and
 done
                //since we need keep sorted, so we need connect left and then append
 root and finally append right
                lefthead = AppendDoublyLinkedList(lefthead, root);
                lefthead = AppendDoublyLinkedList(lefthead, righthead);
                return lefthead;
        }

        public Node AppendDoublyLinkedList(Node head1, Node head2)
        {
                //special case
                if (head1 == null) return head2;
                if (head2 == null) return head1;
                //how to connect two doubly-linked list together?
                //for every list, we need to know head and tail. we already have heads,
 and tail is easy to get since they are doubly-linked list
                int tail1 = head1.Left;
                int tail2 = head2.Left;

                //for tail1 and head2, change left or right;
                tail1.Right = head2;
                head2.Left = tail1;
                //for head1 and tail2, change left or right
                head1.Left = tail2;
                tail2.Right = head1;
                return head1;
        }

 108.Write a function to tell if two line segments intersect or not
    (use cross product, theory here: http://stackoverflow.com/questions/563198/how-do-
 you-detect-where-two-line-segments-intersect/1201356#565282)
        //p + t r = q + u s  (p is point A, q is point C, r is point B - point A, s is
 point D - point C)
        //cross product s on both side (s x s = 0 since a x b = |a|*|b| * sin@) => t =
 (q - p) ¡Á s / (r ¡Á s)
        //cross product r on both side => u = (q - p) ¡Á r / (r ¡Á s)
        //if p - q = 0, means colinear, may has intersect but may not, easy to know,
 add one more condition
        //otherwise, if rxs = 0 means parallel return false;
        //else if 0<=u<=1 and 0<=t<=1, then return true

        // Determines if the lines AB and CD intersect.
        static bool LinesIntersect(PointF A, PointF B, PointF C, PointF D)
        {
                PointF CmP = new PointF(C.X - A.X, C.Y - A.Y); //q - p
                PointF r = new PointF(B.X - A.X, B.Y - A.Y); //r
                PointF s = new PointF(D.X - C.X, D.Y - C.Y); //s

                float CmPxr = CmP.X * r.Y - CmP.Y * r.X;
                float CmPxs = CmP.X * s.Y - CmP.Y * s.X;
                float rxs = r.X * s.Y - r.Y * s.X;

                if (CmPxr == 0f)
                {
                                // Lines are collinear, and so intersect if they have
 any overlap

                                return ((C.X - A.X < 0f) != (C.X - B.X < 0f))
                                                || ((C.Y - A.Y < 0f) != (C.Y - B.Y <
 0f));
```

```
                }

                if (rxs == 0f)
                                return false; // Lines are parallel.

                float rxsr = 1f / rxs;
                float t = CmPxs * rxsr;
                float u = CmPxr * rxsr;

                return (t >= 0f) && (t <= 1f) && (u >= 0f) && (u <= 1f);
        }
```

109.Find an algorithm to find the largest sum continuous subarray in an array of integers. (Kadane's Algorithm)

110.Write a function to prettify Json objects
    (use recursion, object nested in object, for string, number, array, just simply print with proper indentation)

```
        function DumpObjectIndented(obj, indent)
        {
          var result = "";
          if (indent == null) indent = "";

          for (var property in obj)
          {
                var value = obj[property];
                if (typeof value == 'string')
                  value = "'" + value + "'";
                else if (typeof value == 'object')
                {
                  if (value instanceof Array)
                  {
                        // Just let JS convert the Array to a string!
                        value = "[ " + value + " ]";
                  }
                  else
                  {
                        // Recursive dump
                        // (replace "  " by "\t" or something else if you prefer)
                        var od = DumpObjectIndented(value, indent + "  ");
                        // If you like { on the same line as the key
                        //value = "{\n" + od + "\n" + indent + "}";
                        // If you prefer { and } to be aligned
                        value = "\n" + indent + "{\n" + od + "\n" + indent + "}";
                  }
                }
                result += indent + "'" + property + "' : " + value + ",\n";
          }
          return result.replace(/,\n$/, "");
        }
```

111.Given a list of n objects, write a function that outputs the minimum set of numbers that sum to at least K. FOLLOW UP: can you beat O(n ln n)?

112.Write a function that computes log2() using sqrt().

113.Given sorted arrays of length n and 2n with n elements each, merge first array into second array.

114.You are given intervals of contiguous integers, like [1, 10), [15, 25), [40, 50), which are non-overlapping and of a fixed size.

Design a data structure to store these intervals and have the operations of insert, delete, and find functions

115.Print out all the permutations of a string

116.Write a function that prints out all subsets of a given set of numbers.

117.Write a function that takes in two binary strings and returns their sum (also a binary string).

118.Design a system to detect typos and provide suggestions to users
119.Design and implement an algorithm that would correct typos: for example, if an extra letter is added, what would you do?

120.Implement a power function to raise a double to an int power, including negative powers.

121.Given a String containing java-script assets, write a parser which will output the String with proper indentation.

122.Implement strstr in C, char* strstr(char *str1, char *str2) find out if str2 is a substring of str1 and return the first occurrence index
(http://leetcode.com/2010/10/implement-strstr-to-find-substring-in.html)

123.Print a binary tree in infix order. Recursive and iterative.
124.Find the n-th smallest element in a binary tree.

125.Given a certain state of an Othelo game board, location on the board, a certain piece to place on the given location, update the board and make the required validations

126.Delete the node with the associated key in the linked list

127.Design the Facebook Credit system which is a application where users can buy/trade virtual currency and can use the virtual currency to purchase Facebook services, like paid apps.
What's the advantage of the design? How would the total credit points of a user be calculated based on my design?

128.Write a code to convert an ASCII representation of a positive integer to it's numeric value.

129.Write a function to take a BST and a value k as input and have it print the kth smallest element in the BST.
130.Write a function that takes a binary tree as input, and have it perform In order traversal - recursive and then iterative
131.Write a function to take two arbitrarily long numbers in the form of Strings and multiply them, returning another String with the product.
132.Write a function to calculate square root of a number
133.Given a binary tree, write a function to find the length of the longest path in the tree.
(use recursion: the longest path may be in left, may be in right, may be through root node, if through root node, the length is maxLeftHeight + maxRightHeight)
(http://www.geeksforgeeks.org/diameter-of-a-binary-tree/)

134.Find the minimum depth of binary search tree (use BFS, find the first leaf node and then get the min depth, don't use recursion, not optimal code)

135.Pascal's Triangle - print a row (a[n, i] = a[n-1, i-1] + a[n-1, i], but the length of a[n] is n, means the nth row only has n elements)

136.Given an array of integers, find the maximum number that can be reached by summing the best possible consecutive subsequence of the array
137.Design a linked list operation that takes a singly-linked list (only forward ptrs, no backward ptrs) as input and reverses the list.
138.Given a 1TB file of serialized 4 byte integers, and 2GB of ram, sort the integers into a resulting 1TB file. My interviewer was very collaborative in
    entertaining various solution ideas until we came up with a combo that would work performantly and reduce the number of passes over the 1TB file and intermediate files

139.Given a matrix print it clockwise from the first element to the very inner element

140.You are going to take some numbers as an input from a file. You need to witer a program to find longest increasing sequence. You should process it as soon as you are
    taking an input. After finishing the last input immediately you should be able to tell the sequence. Input: 1 5 3 4 6 4 Output: 3 4 6 (longest increasing sequence)
        (LIS problem)

141.A file contains 10 billions of Strings and need to find duplicate Strings. You have N number of systems available. How will you find duplicates?
   (http://stackoverflow.com/questions/3897295/find-duplicate-strings-in-a-large-file)
   Split the file into N pieces. On each machine, load as much of the piece into memory as you can, and sort the strings. Write these chunks to mass storage on that machine.
   On each machine, merge the chunks into a single stream, and then merge the stream from each machine into a stream that contains all of the strings in sorted order.
   Compare each string with the previous. If they are the same, it is a duplicate.

142.First question: for a random-ordered bucket of numbers 1 through 3000 with one number missing, how would you detect which number is missing?
    (if only one missing, no duplicates, use sum of (1...3000) - sum of array, if has duplicates, may use bitmap,)

143.Please write a program to merge 2 sorted arrays (normal way do it)
144.Merge an array of size n into another array of size m+n, only m elements in this array (solution: move m elements to end first, then start from index n t0 m+n, another one just from 0 to n-1)
145.Each key on the telephone represents a list of letters. Given a telephone number, please write a program to output all the possible strings the telephone number represents.

146.Some questions on graph theory and then I was asked to write a function to check if a graph was bipartite.
147.Given two binary trees, return true if they have same elements (irrespective of tree structure)
    (the same as compare if anagrams, traversal two tree, sort, compare if the same.
        or traversal the first, build hashtable <key, count>, then traversal the second, if found, count--, if count=0, remove this key, lastly check if hashtable.count == 0)

148.Given a string, remove all chars from the string that are present in say another string called filter. (din't get it?)

147.Implement stack using a queue
148.Print out all combinations of k numbers out of 1...N e.g. when k = 2, n = 4 Print out 12, 13, 14, 23, 24, 34

149.25 racehorses, no stopwatch. 5 tracks. Figure out the top three fastest horses in the fewest number of races.
    (solution: 7 races)
        We can do it in seven races, we'll call them races A-F. For notation, we'll say that the Nth horse in race X is called X.N.
        Races A-E: Divide the horses into 5 groups of 5 each such that each horse races

only once.
        We can eliminate the slowest 2 horses in each of the five races because there are definitely 3 horses faster in each case. As a result, we eliminate 5x2 = 10 horses: {A.4,A.5,B.4,B.5,C.4,C.5,D.4,D.5,E.4,E.5}
        Race F: Race the fastest horses in each race A-E: {A.1, B.1, C.1, D.1, E.1}. To simplify notation, we'll label F.1 as horse A.1, F.2 as horse B.1, and so forth.
        That means the winner of this race is A.1, and it is the fastest horse of all. We don't have to race A.1 anymore.
        We can eliminate D.1 and E.1 = 2 horses. Because they are not in the top 3.
        As a result, we know that all remaining horses from D and E are eliminated. This is D.2,D.3,E.2,E.3 = 4 horses.
        We know that A.1,B.1, and B.2 are all faster than B.3 (and similar for C.3) so they are not in the top 3.We can eliminate B.3 and C.3 = 2 horses.
        Finally, we know that A.1 is faster than B.1, which is faster than C.1, and thus C.2, so we can remove C.2 = 1 horse.
        Race G: We have removed 19 horses from competition and are sure that A.1 is the fastest horse of them all. This leaves just 5 horses: {A.2,A.3,B.1,B.2,C.1}.
        We race them and select the top 2 to join A.1 as the top 3 fastest horses.

150.Write a list class where the only data structure available is a stack

151.Suppose you have a matrix of numbers. How can you easily compute the sum of any rectangle (i.e. a range [row_start, row_end, col_start, col_end]) of those numbers? How would you code this?
    (use dp? should try real code)

152.Given an array, print the largest subarray that has elements in an increasing order
153.How do you find sequences of consequtive integers in a list that add to a particular number
154.Given a score S, and individual points p1,p2,...,pn. give all combinations of p that add up to s
155.Given a set of characters, print out all possible permutations.
156.Given a binary tree, print out the elements in order. Without recursion.
157.Find the first letter in a string that does not have a pair.
158.Write a function that takes in an integer and returns the number of ones set in the binary representation.
159.Given a large string (haystack), find a substring (needle) on it.
160.Given a list of strings, for each string, find if it has an anagram in the list. (group anagrams)

161.How to implement a DST (daylight saving time)