INTERVIEW PREPARATION – QUESTIONNAIRE

- Ashay Raut

Let me tell you something you already know. The world ain't all sunshine and rainbows. It's a very mean and nasty place and I don't care how tough you are it will beat you to your knees and keep you there permanently if you let it. You, me, or nobody is gonna hit as hard as life. But it ain't about how hard ya hit. It's about how hard you can get it and keep moving forward. How much you can take and keep moving forward. That's how winning is done! Now if you know what you're worth then go out and get what you're worth. But ya gotta be willing to take the hits, and not pointing fingers saying you ain't where you wanna be because of him, or her, or anybody! Cowards do that and that ain't you! You're better than that!

Sylvestor Stalone, Rocky

Maths:

1. Generate power set of given set

```
int A[] = {1,2,3,4,5};
int N = 5;
int Total = 1 << N;
for ( int i = 0; i < Total; i++ ) {
   for ( int j = 0; j < N; j++) {
      if ( (i >> j) & 1 )
           cout << A[j];
      }
  cout << endl;
}
```

2. Find if the given expression contains redundant parantheses. ex :if expr = a+(b*c) , print false, if expr = a+((b*c)), print true.

3. Given a set of intervals like 5-10, 5-10, 8-12, 9-15. Find the ith smallest number in these intervals.

   for eg:-Suppose we have intervals like 5-10, 8-12. Then total numbers in these two intervals would be: {5,6,7,8,8,9,9,10,10,11,12}
   So, 1st smallest number: 5
   4th smallest number: 8
   5th smallest number: 8

4. Add two numbers without using + operator.

5. Negate a number without using just +operator. Eg: negate(5) = -5 ; negate(-5)=5.

6. Find number smallest number K , if exists, such that product of its digits is N. Eg: N = 6, smallest number is k=16(1*6=6) and not k=23(2*3=6).

```
int getnumber(int n) {
   if(n==0|| n==1)
    int i=9;
    int num=0;
    int pow=1;
    while(i>;0)   {
       while(n%i==0 && i>;1)   {
         num=i*pow + num;
         pow=pow*10;    n=n/i;
     }
 i--;
}
 if(n!=1)       return -1;
else  return num;
```

7. How to find the smallest number with just 0 and 7 which is divided by a given number?

8. Count the trailing zeros in factorial of a number N? (5! Has number of 0's = 1)

9. Check if the bit representation of a number is palindrome or not? Eg . 9 => 1001 > is palindrome. Function

```
boolean isPalindrome(int anumber) {
 uint numberOfBits = sizeof(uint)*8;
for(uint i=1; i<= numberOfBits/2; ++i) {
    if( ((aNumber & (1<<(numberOfBits-i)))? 1:0) == ((aNumber & (1<<(i-1)))? 1:0))   {
                                    continue;     }
     else {
                        return false;
          }

}
     return true;
}
```

10. Generate random number generator (mod 7) using (mod 5). Understand the trick then you can generate any random number generator using a given generator.

```
int rand7() {
  int vals[5][5] = {
      { 1, 2, 3, 4, 5 },
      { 6, 7, 1, 2, 3 },
      { 4, 5, 6, 7, 1 },
     { 2,  3, 4, 5, 6 },
```

```
        { 7, 0, 0, 0, 0 }    };
int result = 0;
while (result == 0)    {
        int i =   rand5();
        int j = rand5();
        result = vals[i-1][j-1];
}
```

11. Get the least number after deleting *k* digits from the input number. For example, if the input number is 24635, the least number is 23 after deleting 3 digits. (Courtesy: http://codercareer.blogspot.in/2013/11/no-48-least-number-after-deleting-digits.html)\

12. Find how many squares are present on a chessboard. http://www.programmerinterview.com/index.php/puzzles/find-number-of-squares-chessboard/

Linked Lists:

13. Reverse a doubly linked list. (Just don't ignore the problem because it sounds simple). Code it and be careful at edge cases.

14. Merge two sorted linked lists (**in-place**) , don't form third linked list.

15. Quick-Sort on Doubly linked list. Also try it on Singly linked list.

16. Reverse singly linked list without using three pointers. You can use only two pointers.
eg: p-> points to head ; q-> points to head->next then,

```
while(q) {
        q = p ^ q ^ (q->next) ^ (q->next=p) ^ (p=q) ;
}
```

At the end, p points to newHead.

17. Loop in linked list. Start of loop, if loop exists.

18. Find middle of singly linked list. Break Circular linked list into two different circular linked lists.

19. Number is represented in linked list such that each digit corresponds to a node in linked list. Add 1 to it. For eg: 1999 is represented as : **(1->9 -> 9 -> 9) + 1** => **(2->0->0->0)**
Extension : Don't reverse linked list.

20. Reverse nodes in linked list problems. Like : k-alternate node reverse **OR** reverse k nodes, then next k nodes and so on..

Arrays:

21. You have an array which has a set of positive and negative numbers, print all the subset sum which is equal to 0.

22. Divide a given array into two subarray (not necessary to be continuous) such that difference between sum of both array is minimum.

23. Find number of inversions in an array. Inversion means a pair where a[i] > a[j] and i <j in O(nlogn).

24. Triplet sum problem. Find all pairs of numbers such that a[i]+a[j]+a[k]=N.

25. A stream of integers is coming. Find median of numbers received till now.
Hint: Two heaps required, one minheap and other one maxheap.

26. Find equilibrium index in array. Its the index i where sum till i from 0^{th} index = sum from ith index to last index.

27. Find the Minimum length Unsorted Subarray, sorting which makes the complete array sorted. Eg: (0 4 12 3 15 16 18 17 19).. Seems like mean length unsorted subarray is from 14 to 17, but its not. Because if you sort that partition you get, (0 4 3 12 15 16 17 18 19) which is not sorted. Think of all cases and find minimum length such that , if we sort that contiguous subset, we get sorted array.
Hint : In above example, once you find shaded part, just get min and max in that shaded part and check if anything in left of that partition is not greater than min and anything on right side of that part is not less than max, so that once you sort that partition, you will get sorted array.

28. Numbers in array are repeated thrice except one number which is present just once. Find the one?
Hint : A number appears three times, each bit (either 0 or 1) also appears three times. If every bit of numbers appearing three times is added, the sum of every bit should be multiple of 3.

29. Write heap functions to build heap, add new element in heap.

30. Find peak element in array. A[i] is peak element, a[i-1] < a[i] > a[i+1].. For corner elements a[0] is peak, if a[0] > a[1] **or** a[length-1] > a[length-2] then a[length-1] is peak. There can be many peaks in array. Find anyone.
No repetition of elements in array. Try it on O(logn).

31. There are 0's and 1's in array. Bring all 0's on odd positions and 1's at even positions. If 1's or 0's are more, leave the remaining as it is after arranging at even/odd positions.
Hint: Keep two pointers. Odd and even position pointers. Both will move +2 steps at a time. Keep moving odd pointer ahead till u get a place where 0 is not present. Do the same for

even pointer. Then exchange values of these two pointers. Continue till anyone of them reaches end of array.

32. Given an array of integers, sort the array according to frequency of elements. For example, if the input array is {2, 3, 2, 4, 5, 12, 2, 3, 3, 3, 12}, then modify the array to {3, 3, 3, 3, 2, 2, 2, 12, 12, 4, 5}.
Hint: Insert elements into maxheap. Condition for maxHeap is frequency of elements. So root will contain element that has maximum no of repetitions.

33. The cost of a stock on each day is given in an array, find the max profit that you can make by buying and selling in those days. For example, if the given array is {100, 180, 260, 310, 40, 535, 695}, the maximum profit can earned by buying on day 0, selling on day 3. Again buy on day 4 and sell on day 6. If the given array of prices is sorted in decreasing order, then profit cannot be earned at all.

34. You are given a one dimensional array that may contain both positive and negative integers, find the sum of contiguous subarray of numbers which has the largest sum. For example, if the given array is {-2, -5, 6, -2, -3, 1, 5, -6}, then the maximum subarray sum is 7. Dont use Kadane.. Try with Divide and conquer approach. Refer geeksForGeeks.org

35. Find maximum difference in indexes. Say, (j-i) is answer if i,j are both indexes such that i<j and A[i]>A[j]; .. This problem is different from the one below..

36. Maximum difference in two elements in arrays such that the larger elements appear on right-side.

37. Suppose array values represent height of histograms. Find largest area rectangle formed by some contiguous histograms.

38. Imp problem: Find number which is repeated more than n/2 times in array where n is length of array. Extension: Now find element which is repeated more than n/4 times, n/8 times and so on.
Hint: http://www.cs.utexas.edu/~moore/best-ideas/mjrty/ This will help you find n/2 times repeated element. To find(n/4 or more)times repeated element, repeat same algorithm but this time ignore the one which is already found as repeated n/2 times.

39. Find number of elements on right side, which are greater in value then current element. Find this for all elements of array. (Hint: Use BST tree).

Stacks and Queues:

40. Simplest and famous one: Design stack for top(),push(),pop() and getMin() operations in O(1) time complexity.
Hint: Two stacks. Beware of edges cases like when last element is popped out.
Extension: How would you design for getMax() operation also.

41. Find the next smallest element on right side for every element a[i] in array a[].

42. Design queue for getMiddle() in O(1) time complexity.
Hint : Think in terms of how to implement queue. How about LinkedList. Can we maintain one additional pointer to middle element always and update it whenever new element is added at rear.

43. Design Queue for getMin() in O(1) time complexity. Hint: Use two queue.

44. Inorder/postorder **iterative** using explicit stack. This problem is important as this technique is required in many problems.

45. Find maximum number in sliding window of k . eg : if array is { 1,5,7,2,1,3,4} and k=3, then first window is {1,5,7} so maximum is 7, print 7, then next window is {5,7,2} , maximum is 7, print 7. So final output is : { 7,7,7,3,4 }
This is very famous problem usually asked in big companies during initial screening rounds.

46. Celebrity problem : There are n number of people in party and only one celebrity. If celebrity is the person, who knows no one but everyone knows him, find who is the celebrity? Given: a function is provided which tells whether 'a' knows 'b' or not? doesAknowB(A,B).
Hint: Push all the elements into stack. Remove top two elements(say A and B), and keep A if B knows A and A does not know B. Remove both A,B is both know each other.

Trees:

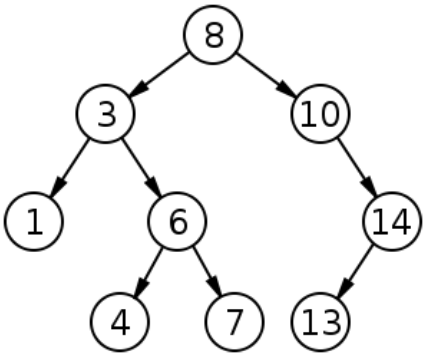
47. Find any two elements in BST such that sum of those two nodes = K.
Hint: Try O(logN). Do 2 iterative inorders, one for left-root-right and other following right-root-left. Add those two values and check cases like <k, >k,==k.

```
inroderRevInorderBoth(root) {
done1, done2  = false;
while(1)    {
    while(done1==false) {
        while(curr1) {
                s1.push(curr1);
               curr1=curr1->left;
            }
     if(s1.empty())
       done1 = true;
     else {
        val1  = s1.pop();
        curr1 = val1->right;
        done1= true;
       }
      }

 while(done2==false) {
   while(curr2) {
       s2.push(curr2);
      curr2=curr2->right;
    }
     if(s2.empty())
       done2 = true;
     else {
       val2  = s2.pop();
      curr2 = val2->left;
     done2= true;
  }
 }
 if(val1->data + val2->data==target)
     return;/*print answer val1, val2*/
  if(val1->data + val2->data < target)
    done1=false;
  else
    done2=false;
 }
}
```

48. Convert a binary tree to BST. Extra space allowed. (Can you maintain same shape of the original tree and still get BST. Answer is no..(http://stackoverflow.com/questions/3531606/convert-binary-tree-bst-maintaining-original-tree-shape)

49. Given a BST, delete a node in BST. (Refer to example below for hint. Node with value 6 is going to be deleted from its original position).

50. Given a BST, and a node in BST, make that node, as new root of the tree. But still maintain the tree as BST after making this node as root. For eg: In tree below, say given node is 6, then change the tree to make 6 as root of tree..
Hint: Take 6 as root, as its on left of original root(8) so, make 8 as right child of 6 and 3 as left child node. Now, there are two cases:
- If 6 is leaf, then no worries at all
- if 6 is internal node, remove 6 and replace it with (either max in left substree or min in right subtree)



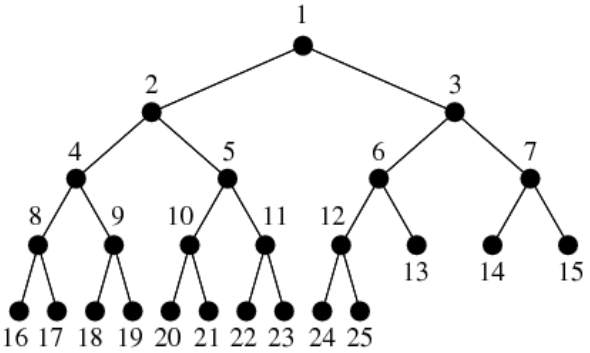
51. Find the inorder successor of BST in O(logN). By using inorder, you can find it in O(n) but then you are not using its BST properties.

```
public class inorderSuccessor {

public static TreeNode iterativeSolution(TreeNode root, TreeNode target) {

        if (root == null || target == null) return null;
            boolean targetFound = false;

        TreeNode currNode = root, candidate=null;
        while (currNode != null) {
                if (currNode.val == target.val) {
                        targetFound = true;
                        currNode = currNode.right;
                }
                else if (currNode.val > target.val) { /* only update candidate here as we r
looking for successor.. it will always be greater.*/
                        candidate = currNode;
                        currNode = currNode.left;
                }
                else {// currNode.val < target.val
                        currNode = currNode.right;
                }
        }
        return targetFound ? candidate : null; // important check to return candidate.
    }

}
```

52. Determine if a tree is a valid BST with no duplicated values. (This means that if the binary tree has a duplicated number it should return "invalid" even if it's an actual BST)

Hint: Check if left subtree or right subtree is invalid or not. Also keep track of prev node visited in inorder traversal and check if value is repeated by comparing previous visited node with current node.

53. Implement a function which returns list of all nodes in a binary tree having a given number of leaves, say k . Also mention complexity. Eg: k=4, ans is {4,5} .. k=2 ans is {8,9,10,11,12,7}
Hint: Think bottom-up, so at every node, getall nodes from left and right subtree and see if its k, then print the node.



54. Replace each node with the sum of all greater nodes in a given BST?
Hint-1 : If converted to Doubly linked list, then you can take prefix sum, starting from tail to head. But this will modify the tree.

Hint-2: Try reverse inorder(right, root,left). Take sum of all nodes in right first, add current node's value to it and pass to left subtree. Total sum returned by left subtree can be passed to root. Recursion is the key.

55. Do **iterative** inorder, preorder **without** using stack. No extra space allowed.
Hint : Morris traversal.

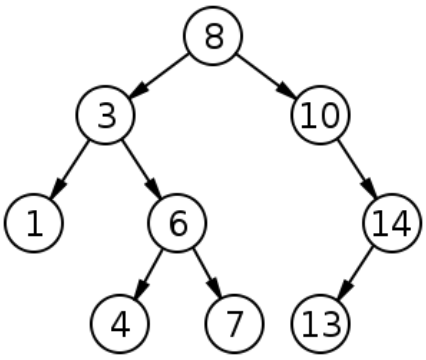
56. Find least common ancestor in tree for given nodes 'a' and 'b'. But time complexity should be O(n) and note its binary tree and not BST.

57. Print all paths in trees (not just root to nodes).
Hint: O(n^2) to get all pairs of nodes and to print path between them, you can find LCA(a,b) between nodes 'a' and 'b' to get path between them, which will take O(n) so, total complexity O(n^3). Can you get better?

58. Find median in BST. There are can be different approaches. Extension to question: Suppose 'K' number of queries are coming in such a way that they ask for i-th smallest element in BST.
Hint: For extension, you can preprocess tree such that, it stores extra information about number of children in left/right subtrees so that, we can traverse tree based on value of 'i'. Eg: if a node has 5 nodes in left subtree and 3 on rightside, and i=7, then at node you know, you have to go in right substree.
So, for 'K' queries, O(logn) will be required to get i-th element.

59. Two nodes in BST are incorrectly placed. So, swap the two nodes such that BST is formed correctly.

60. Form BST from given preorder traversal output.
eg:
Preorder given : 8,3,1,6,4,7,10,14,13 ... Think recursion, 0^{th} element is root, then find elements that qualify for left subtree and right subtree and go recursive to get actual trees.
Output:



61. Find distance between two keys in Binary tree. (not BST).
Hint: LeastCommonAncestor

62. Given a binary tree, a target node in the binary tree, and an integer value k, print all the nodes that are at distance k from the given target node. No parent pointers are available.
Hint: Refer geeksForGeeks.org

Given a binary tree where each node has an additional pointer filed *'sibling'* (in addition to the *'left'* & *'right'* fields), connect all the siblings. Note, the tree may not be a complete binary tree.
Hint-1: Do level order traversal and previous node is connected to current node in level order traversal.
Hint-2: For every node, (if right child exists),connect its left-child to right-child, if not, connect left child to (find node by traversing, sibling pointer of current-node, and finding appropriate next sibling for left child).
For right child, find sibling by traversing, sibling of current node and finding required node.
eg. In example tree of question 60, at node 8, connect 3's sibling pointer to 10.
At 3, connect 1's sibling to 6, and for 6, traverse sibling of 3 which is 10, and get 10's right child(14) as left is not present. Attach sibling of 6 as 14. And so on.

Strings:

63. Find longest repeating substring in string. Eg; abcdsabcaewcabcad ..Here ans is 'abc'.
Hint: http://en.wikipedia.org/wiki/Longest_repeated_substring_problem

64. Find longest palindrome substring in O(n2) without extra space.
Hint: No dynamic programming required.
- For odd length palindromes, From each element, moves both directions left+right and check if palindrome is formed and record length if maximum.
- Also, for even length palindromes, start from between every two elements and go both directions till palindrome property is formed.

65. Famous problem for interviews: Given a string 'str'and some other string 'str2', find smallest substring in str which will contain all characters of 'str2' even if they are in any order.
eg: str="bzdsdaddwcfaobwww" str2="abc". Highlighted part is answer.

66. KMP search : Find if given 'pattern ' is present in given string "str", if present return the index of start of pattern in str. For eg: str="asdqeaicicaiciciaaw", pattern='icici', then ans is 11.
One of the Important questions, understand KMP search properly as its frequently asked.

67. Design String class of our own. Be careful to design understanding all properties of Java String class like immutable. (Hint: See how to make class immutable and check Java String class properties)

68. There is a dictionary of billion words and there is one method provided "String getWord(int index);". We can give it index and it will return the String on that index.
Hint: Use Binary search if end index is known. Or else try Fibonacci search to reach a point where string at that index is greater than given word.

69. String compression**(in-place)**. O(n) required. For given string compress using following logic:
case1 : input string : aaawwwbbbcc
compressed output: a3w3b3c2
case 2: input string : acccbww
compressed output: a1c3b1w2

Assume that every input string has infinite amount of memory.
Hint: Count number of places where no repetition is present but don't insert 1 now, just count.
- For characters where repetition is present, replace duplicate characters by digits. At last, let compressed string be called str(as its in-place).
- To insert 1's at places where no repetition is there, start copying each char from end of str to str.length+count position. This is new length of string.
- While copying for characters for which number is not present in-front of them, it means, they appeared once, so insert 1 and then the character.
- Eg: str=acccb After transformation, str=ac3b and new_length=4 and count=2
So, keep pointer at end of str, at 'b', and copy it to new_length position in str.
But we observe that for 'b' , no number appeared before it so, its not repeated, so we insert 1 at that new position.
Then, we get number, so for numbers, just copy them to new postion as it is. (new position - 1) is to be done every time.
Now, for c, we saw a number just before 'c' so, it means, c was repeated so copy 'c' as it is.

Now, comes ‘a’ and we see no number associated with it, so insert ‘1’ and then ‘a’.
So final answer : a1c3b1

Dynamic Programming(Only frequently asked):

70. Find length of longest series questions :
- Longest common subsequence in two strings
- Longest common substrings in two strings
- Longest increasing subsequence in O(nlogn)
  Refer : http://www.algorithmist.com/index.php/Longest_Increasing_Subsequence.cpp

71. You are given an array A with elements 0 to n-1, numbers can be repeated in the array. Create n sets where
S[i]={a[i],a[a[i]],a[a[a[i]]]…}. Set has all elements unique. Find the size of the largest set.
Input:
First line contains n, size of the array. n<1000
Next lines contains n numbers, each element of the array
Output
Prints one number: Size of the largest set.
Sample Test Case:
Input: {3,1,2,0} Output: 2
Explanation:
Four possible sets are
{3,0},{1},{2}{0,3}
Refer: http://www.careercup.com/question?id=4724898538717184

72. Find optimal binary search tree that can be formed from given sorted array of values.
Better explanation for dynamic programming given at :
http://orion.lcg.ufrj.br/Dr.Dobbs/books/book2/algo032a.htm

73. Check if there exists, **non-contiguous** subset in array such that sum of subset is exactly total_sum/2.
Where total_sum = sum of all elements in array.

74. Mostly asked in written test of Amazon: Given coins of certain denominations and their infinite supply, find minimum number of coins required to get required ‘K’ amount of money.
Variation: There can be only one coin of each denomination, now tell if K sum amount can be formed or not.

75. Word break problem. Eg: Iamlive, this string can be broken,i am live, which are three valid words in dictionary. Figure out , if a string can be broken down into valid words or not? Function written should just return true/false.

76. A frog can jump 1 or 2 steps at a time. In how many different ways can he reach distance K.
Hint: Is Fibonacci related to this problem?

77. There are N houses in a row. Each house can be painted in either Red, Green or Blue color. The cost of colouring each house in each of the colours is different. Find the colour of each house such that no two adjacent houses have the same colour and the total cost of colouring all the houses is minimum.
Hint:

- If you are calculating cost of colouring till ith house and you know the minimum cost of colouring the (i-1)th house and color of (i-1)th house, then you can color ith house to keep cost minimum. There will always be two colors by which you can paint ith house except 0^{th} house(0th you can color in 3 ways). So at every house, there will be two values and you need to consider both of them when calculating the cost of next one. At last house, we can paint it using any two colours again, choose the one which gives minimum cost.

- Point is only at last house, you can choose only one colour, the colour that gives minimum cost. At all other houses, you have to maintain two values for two colors, you are giving them. Because, if at every house, you keep choosing only the color that gives minimum that is greedy approach and may not always give correct answer.

78. Edit distance problem. (http://en.wikipedia.org/wiki/Edit_distance)
Solution at http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/

79. Matrix chain multiplication problem. Very famous problem.

80. You are given a boolean expression consisting of a string of the symbols 'true', 'false', 'and', 'or', and 'xor'. Count the number of ways to parenthesize the expression such that it will evaluate to true. For example, there are 2 ways to parenthesize 'true and false xor true' such that it evaluates to true.
Hint: http://people.cs.clemson.edu/~bcdean/dp_practice/dp_9.swf

Miscellaneous problems:

81. Find median,if K sorted arrays of length N are merged.
Hint: MinHeap. Take min element from each array and add to minHeap. Remove min element from root of heap and insert new element into heap from the array to which root belonged. Keep on doing this, till you get to median. (Median => if K*N is odd, then median is (K*N)/2 + 1 th element, and if (K*N) is even, then ((K*N)/2 + ((K*N)/2)+1))/2 th element.) Check this condition once.

82. Find top 10 frequent words in a file with millions of words.
Hint: Trie and minheap. Size of heap will be 10. Insert word into trie, the last node containing last character of word will have count which will incremented by 1 , everytime word is inserted. If count > (the minimum value in minheap(i.e. root's count) then remove root and insert this word into heap. MinHeap is arranged based on number of occurances a word had till now.
Food for thought: Can we used doubly linked list instead of minHeap to reduce time complexity.
Extension : Instead of top 10 words, it can top 10%. In that case, the minheap size will change such that everytime it will store 10%(high-frequency words) of the words read till now.

83. Implement Least Recently used cache. Write code for it too.

84. Find maximum sum rectangle in matrix of size m*n with 1's and 0's.
Hint: You will need Kadane algorithm to reduce time complexity from O(n^4) to O(n^3). Refer geeksforgeeks.org for solution.

85. Given a number find the next smallest palindrome.
eg: 123 -> 131, 991 -> 1001, 121 -> 121 itself.
Hint:
- There are two cases, whether the number of digits in the number is odd or even. We'll start with analyzing the odd case.
- Let's say the number is ABCDE, the smallest possible palindrome we can obtain from this number is ABCBA, which is formed by mirroring the number around its center from left to right (copying the left half onto the right in reverse order).
- This number is always a palindrome, but it may not be greater than the original number. If it is then we simply return the number, if not we increase the number. But which digit to increment?
- We don't want to increment the digits on the left half because they increase the number substantially, and we're looking for the smallest palindrome larger than the number.
- The digits on the right half increase the value of the number much less, but then to keep the number palindrome we'll have to increment the digits on the left half as well, which will again result in a large increase. So, we increment the digit just in the middle by 1, which corresponds to adding 100 in this case.
- This way the number stays a palindrome, and the resulting number is always larger than the original number.

Courtesy: Ardent (http://www.ardendertat.com/2011/12/01/programming-interview-questions-19-find-next-palindrome-number/)

86. Backtracking problems: N-Queens problem. Place N queens in such a way that, they can't kill each other.

87. One knight is placed at some position Y in chessboard, find in how many minimum moves it will move to some given position X in chessboard. By position X or Y means, it will have row and column number.

88. Print all permutations of number in sorted order formed by last digits. Eg: number=123, then numbers formed are: 123,132,213,231,312,321.
Hint:
To get next number which is greater than given number.
1. Start from rightmost digit and go left till you find ith element such that, digit_at_i < digit_at_i+1.
2. Now go find on right-side of ith element, find the element(say at position 'j') that is greater than ith element but smallest among all those elements which are greater and on right side of ith element.
3. Swap the digits at position 'i' and 'j'.
4. Sort digits from i+1th position to end.

89. Given a text file and a word, find the positions that the word occurs in the file. You'll be asked to find the positions of many words in the same file.
Hint: Do some precomputation which will help to reduce complexity of getting position of word in text. So kind-of inverted index (word -> position mapping). Choose datastructure. It can be hashtable or trie(at whose end node will have List<Integer> indexes where the word occurred.)

90. Given a matrix of integers and coordinates of a rectangular region within the matrix, find the sum of numbers falling inside the rectangle. Our program will be called multiple times with different rectangular regions from the same matrix.
Hint: Refer (http://www.ardendertat.com/2011/09/20/programming-interview-questions-2-matrix-region-sum/)

91. Implement Trie operations -> insertion, deletion and lookup operations.

92. Stream of 0's and 1's is coming, find whether the number formed together by this binary bits is multiple of 3 or not. The stream will keep sending 1 bit at a time. At every bit, tell the new number formed is multiple of 3 or not.
1) Get count of all set bits at odd positions (For 23 it's 3).
2) Get count of all set bits at even positions (For 23 it's 1).
3) If difference of above two counts is a multiple of 3 then number is also a multiple of 3.

93. From interview standpoint, always know and have understanding about problems like Reader-writers problem and Producer-consumer problem. Also know their solutions using semaphores or monitors.

94. Train station timings are given. That is, at a station arrival and departure times of trains is given. Find minimum number of platforms to be constructed at that train station so that trains at similar interval range can be accommodated.

Eg:

Train 1 Arrival:9:30am Depart: 10:30am

Train 2 Arrival:9:45am Depart: 10:10am

Train 3 Arrival:10:35am Depart: 11:30am

Train 4 Arrival:11:55am Depart: 12:10pm

Train 5 Arrival:11:50am Depart: 12:15pm

Ans: 2 (Train 1,2 and Trains 4,5 overlap, so two platforms are required)

Hint: The required platforms is the number of trains whose timings overlap. If none overlaps, then we can do with one platform.

- Sort all the timings including arrival and departure timings by keeping them in one single array. And maintain that a particular timing is for arrival and departure. So, if n trains are present, you sort an array with 2n timings(arrival and departure time for each train).
- Once you sort these 2n values, start from 0^{th} index and increment count as you encounter timing value(marked as 'arrival') and decrement count for timing (marked as 'departure').
- Also, maintain the maximum value obtained by count somewhere because at last this max value achieved by count is the answer.

Design Questions

95. Design an elevator system.
Hint: Think in terms of algorithm to be used for elevator movement. Different usecases. Focus on classes and interaction between them.

96. Design spell-checker for MS-word.
Hint: Think about data-structure like trie or ternary search tree.

97. Design Chess game. Define classes and interactions.

98. Design web-application that will accept pincode as input and will return list of departmental stores in that pincode.
Interviewer will not say anything, after this line, he will accept you to come up with databse design and what data you will store in db to figure out which stores lie in a particular pincode.
Hint: Also, don't forget its web-application, so do focus on caching, multiple servers, load balancing for additional points ☺

99. Implement a T9 dictionary. Its keypad of mobile. When you press certain numbers, you need to suggest words that are possibly formed from characters represented by those numbers.

100. Imagine you're designing a Web Service for a phone application that returns a list of suggested Words that may complete a given string the user types. For example, if the user writes "ap", a list of suggested words may contain ["apple", "application", ,aptitude",...]Assume English only words and no misspelling.

"It's hard to beat a person who never gives up."..Babe Ruth