100 Questions Challenge

Semester: III Division: C Year: 2019-20

While you see the hundred, and you think of hundred,
In time and space, you are away from a hundred!

Note: Repeated questions will not be evaluated. Unnumbered will not be too.
Total Challenge Points: 3600 Threshold you need to cross: 2700

Set A – Creative Corner

Write a C Program for the following poems **[80 Points]**

1. Data, data, all I see,
Integer data, data of sea,
Biggest of all is what I need,
Rest of them is just a weed.

2. Take a number, in one to ten,
Take another, random one,
Play with them, as you like,
Give me unique, will stop the strike.

3. I stand out, see the sky,
Never-ending, is it a lie?
The chance I get is until five,
Just don't ask me what and why!

4. Strings are coming one by one,
I keep counting vowels ten,
After ten, I go to my den,
I retire and get my pension.

5. Push, Pop, Peek, and Print,
The top gives an entire hint.
The first will be the last out,
Stack it is, no doubt!

6. Rear to attach,
Front to detach,
First in First out is the new,
Enqueue, Dequeue, It's the Queue.

7. Malloc makes a call,
Pointer makes a move,
Node houses them all,
Thus the list is born!

8. Give the key,
Get the address.
Makes the data,
faster access.

9. Answer the following: Each answer will fetch 1 point. Creative and logical answers will fetch 5 points. Not answering will fetch -2 marks. **[150 Points]**

9.1 Reversing a palindrome string using the stack, Hilarious or Geeky?
9.2 What do you get when you peek into an empty stack?
9.3 To 'understand recursion,' you need to understand 'recursion.' Understand?
9.4 If a pointer goes dangling, what did the free() do?
9.5 "Being Key Address," whose t-shirt quote would it be?
9.6 Why was rear index unhappy?
9.7 Where do you scatter when you collide?
9.8 A node had two nulls. Was it too many?
9.9 What did it trie?
9.10 Who left the pointers out? Who? Who? Who?
9.11 Who is the major supporter of malloc()?
9.12 Who would stand against 'static' in elections?
9.13. Who terminates a struct definition?
9.14 Why does pointer consider itself to be the coolest dude?
9.15 Why does not tree watch Hindi movies?
9.16 If stackie and queueie were twins, what would be their differing features?
9.17 The stack is growing at one end and heap at another. What happens next?
9.18 Is data structure male or female? Why?
9.19. Why are strings, array?

9.20. If that's a chair and scope is a table, what's that?
9.21 If a DFS was run for every iteration of DFS, what would you finally get?
9.22 A level order traversal wasn't happy at all. What day was it?
9.23 A DFS post and pre-order numbering were used on a DFS forest. The end number turned out to be 23. Anything weird?
9.24 How are BST and AVL related?
9.25 Why wasn't BST allowed to enter inside a party but whereas AVL and 2-3 Trees were allowed?
9.26 Why was the root of the tree scared?
9.27 Why are single rotations of an AVL tree prouder than double rotations?
9.28 2-3…. what comes next?
9.29 Who is more organized? Inorder? Preorder? Or Postorder?
9.30 How are binary and binary search tree related?

10. Write as many examples you can write for stacks. The example you write, if also written by other team, it will fetch you -5 points. If unique, will fetch you 10 points. **[100 Points]**

11. Write as many examples you can write for queues. The example you write, if also written by other team, it will fetch you -5 points. If unique, will fetch you 10 points. **[100 Points]**

12. Design a logo for the following. **[100 Points]**
If you were to design a metaphor that would stand as an icon, then what would be it? Like the one you see for Recycle Bin, My Computer etc.
a. Tree b. Algorithm c. Recursion d. Orders of Growth e. Hashing

13. Talking to a kid! **[20 Points]**
How do you explain to your 5 year old nephew who desperately wants to know 'what are data structures and algorithms'? Now, don't discourage the kid by saying it's an Engineer's headache!

14. Talking to grandma! **[20 Points]**
How do you explain to your 70 year old grandmother who desperately wants to know 'what are data structures and algorithms'?

15. With the help of suitable code snippets, **[20 Points]**
Prove That: "Queue is NO more exactly a First In First Out data structure"

Set B – The Usual's.

Solve the following Questions. Each carries 20 Points. **[400 Points]**
16. Define the following:

a. Strictly Binary Tree	e. Almost Complete Binary Tree	i. Ternary Tree
b. Direct Address Table	f. Hash	j. Trie
c. N-ary Tree	g. DFS	
d. Subset Sum Problem	h. Quadratic Probing	

17. Suppose we wish to search a linked list of length **n**, where each element contains a key **k** long with a hash function **h(k)**. Each key is a long character string. How might we take advantage of the hash values when searching the list for an element with a given key?

18. A linear probing has a hash function of the form: $\mathbf{h(k, i) = (h'(k) + i) \bmod m}$ and a quadratic probing has a hash function of the form: $\mathbf{h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m}$. Linear probing suffers from a problem known as primary clustering and quadratic probing from secondary clustering. Discuss.

19. Suppose that we store **n** keys in a hash table of size $\mathbf{m = n^2}$ using a hash function **h** randomly chosen from a universal class of hash functions. Then, the probability is less than ½ that there are collisions. Argue. Hint: Apply Markov inequality.

20. Write a program for Towers of Hanoi problem. Show the manual tracing for 2 and 3 disk problem.
21. Iteration is better than recursion. Comment with three to four justifying statement.

22. Explain the below algorithm:

```
HASH-INSERT(T, k)
i = 0
repeat
   j = h(k, i)
   if T[j] == NIL
      T[j] = k
      return j
   else  i = i + 1
until i == m
error "hash table overflow"
```

23. Optimize the code given below. Compact the code to fewer lines.

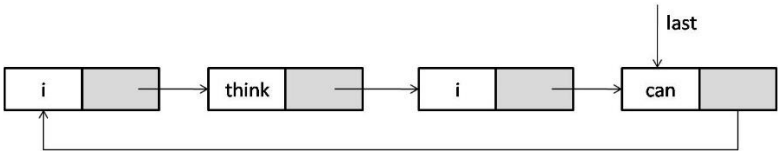
```
Void displayQueueElements(Queue *q) {
   int i = 0;
   if(q-> front > q->rear) {
      printf("QUEUE EMPTY\n");
      return;
   }
   else {
      printf("QUEUE ELEMENTS ARE\n");
      for(i=q->front; i<=q->rear; i++) {
         printf("%d\n", q->items[i]);
         return;
      }
   }
}
```

24. Define and explain backtracking in layman terms using an example.

25. Following are the keys and addresses generated using a hash function. Demonstrate on how the keys would be hashed if the hashing technique used was chained progressive overflow.

Key	Address Generated
Rachel	43
Monica	41
Phoebe	43
Joey	43
Chandler	41
Ross	43

26. Consider the circular list given below with string data



Write a function which will display the output in following fashion:
i think i can
think i can
i can
can

27. Write a function for:
// **Function Name:** summon
// **Input Params**: base address of string
// **Return Type:** base address of summoned string
// **Description:** A magician wants to generate summoning charms.
// For input string "firebolt", the function should produce "summon firebolt". Prefix the passed
// string with 'summon' and return it. For a successful magic, pointer operations are mandatory and
// usage of inbuilt string handling functions are restricted.

28. The 'N' queen problem
Using **state space tree** prove that:

- There is no solution for a 2 queen problem
- There are multiple solutions for a 4 queen problem

29. Explain the various collision resolution techniques.
30. With code snippets explain iteration and recursion. You need to pick one function, write both recursive and iterative code and then explain.
31. With a help of a neat diagram explain the difference between call by value and call by reference.
32. Write a function of find the height of a binary tree.
33. Write a note on algorithm analysis framework.
34. Explain Asymptotic notations.
35. Write a note on application of trees.

Set C – List Challenges. Mind it!

Write functions for following list tasks: **[440 Points]**

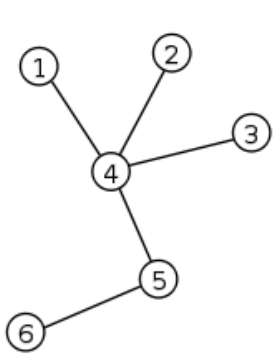
36. You are given a Linked List and a number K. You have to reverse it in the groups of K
Ex : [1] -> [2] -> [3] -> [4] -> [5] -> null, K = 3
output: [3] -> [2] -> [1] -> [5] -> [4] -> null
37. Write a function to get the intersection point of two Linked Lists.
38. Given two linked lists find if they are making a shape of 'Y' or a shape of 'V'.
39. Write a function to check if a singly linked list is a palindrome.
40. Construct a linked list from 2D matrix.
41. Write a code to merge two linked lists.
42. Write a function to perform union and intersection on two singly linked list.
43. Remove duplicates from a sorted linked list.
44. Writ**e** a function to reverse a list.
45. Implement a function to get the middle node of the list.
46. Consider sorted singly linked list having following nodes.
10->30->50->70->NULL
You are given pointer to node 50 and a new node having value 40. Can you inserte node 40 correctly in the list maintaining the ascending order?
47. Given number k, for Single linked list, skip k nodes and then reverse k nodes, till the end.
48. Reverse a doubly linked list without using any extra space.
49. Write a function to print alternate nodes in a linked list.
50. Implement sorting on a singly linked list.
51. Write a function to detect a loop in a singly linked list.
52. Append the last 'n' nodes of linked list to the beginning.
53. A list has repeating data items. Find the count of repeat of each data item in the list.
54. Find the max and min elements in the linked list.
55. Accept an integer value from the user and convert each digit into a node of list.
Example: Input: 4872
Output List: 4->8->7->2->NULL
56. Write a function to insert a node at nth position in a doubly linked list.
57. Write a function to concatenate two given lists.

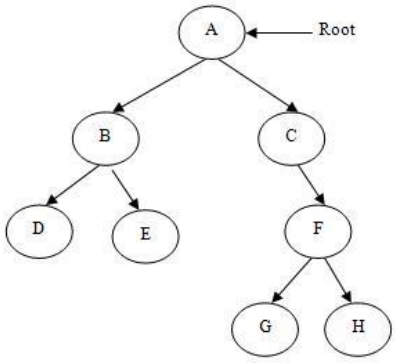
Set D – The Randoms

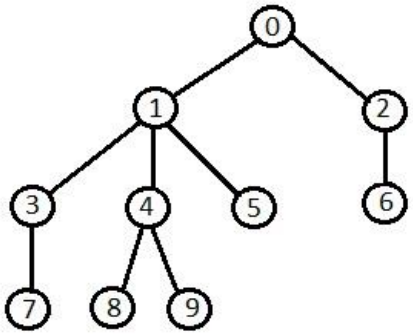
58. For the numbers: 35, 72, 98, 56, 33, 89, 62 **[50 Points]**
a. Create a Binary search tree, AVL tree and 2-3 tree
b. Write down the table populating the number of comparisons required to search each element in all the three cases

59. Construct a BST, 2-3 tree and AVL tree for the given inputs: **[60 Points]**
a. T E C H N O L O G Y
b. 89, 21, 13, 17, 96, 45, 37, 41, 83, 67

60. Write the three tree traversals for given below trees: **[60 Points]**







61. Consider the following algorithm: **[60 Points]**

```
ALGORITHM Min1(A[]0… n-1)
if n = 1 return A[0]
else temp ← Min1(A[0… n-2])
  if temp <= A[n-1]
     return temp
  else
     return A[n-1]
```

a. What does the algorithm compute?
b. Which is the Basic Operation?
c. Set up a recurrence relation and solve it.
d. Which class of problem does the algorithm belong?
e. Can you make the algorithm efficient?

62. You are given an array with duplicates. You have to sort the array with decreasing frequency of elements. If two elements have the same frequency, sort them by their actual value in increasing order. **[30 Points]**
Ex: [2 3 5 3 7 9 5 3 7] Output: [3 3 3 5 5 7 7 2 9]

63. Given a character array as input. Array contains only three types of characters 'R', 'G' and 'B'. Sort the array such that all 'R's comes before 'G's and all 'G's comes before 'B's. Constraint :- No extra space allowed (except O(1) space variables) and minimize the time complexity. You can only traverse the array once. **[50 Points]**

64. A hash function h defined h(key)=key mod 7, with linear probing, is used to insert the keys 44, 45, 79, 55, 91, 18, 63 into a table indexed from 0 to 6. What will be the location of key 18? **[50 Points]**

65. Given an array containing sequence of bits (0 or 1), you have to sort this array in the ascending order i.e. all 0' in first part of array followed by all 1's. The constraints is that you can swap only the adjacent elements in the array. Find the minimum number of swaps required to sort the given input array. Example: Given the array (0,0,1,0,1,0,1,1) the minimum number of swaps is 3. **[60 Points]**

66. You are given an array which contains either 1 or 0 , and they are in sorted order Ex. a [] = { 1,1,1,1,0,0,0}. Devise an efficient method to count the number of 0's and 1's. **[50 Points]**

67. Explain the deletion cases in BST with example and code. **[50 Points]**

PH

68. Write a recursive code for binary search and perform efficiency analysis **[30 Points]**

69. Construct a Trie with mentioned operations as below: **[50 Points]**

Insert:

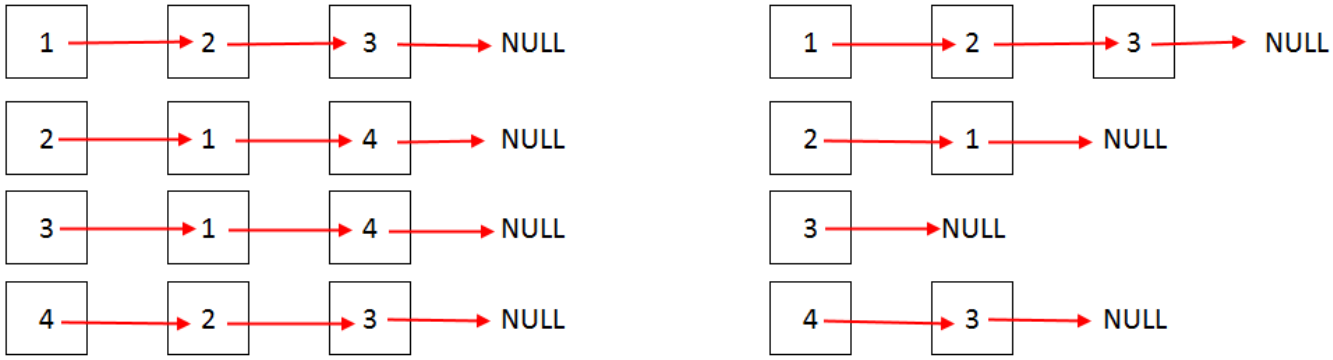
P, 1 pr, 2 PrE, 3 pref, 4

PreFi, 5 PrefiX, 6

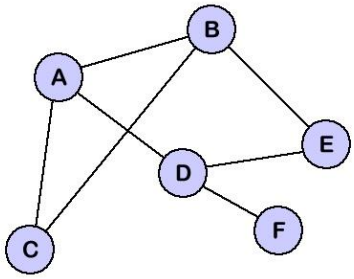
Search: pr and prefix

Delete: P and pref

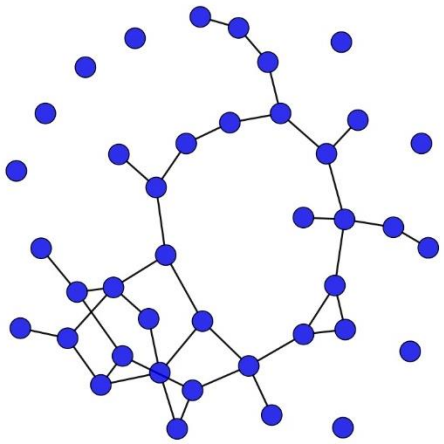
70. Write DFS and BFS for given two graphs. Consider 1 as source vertex. **[40 Points]**



71. Perform DFS and BFS for given graph starting from all the vertices. **[60 Points]**



72. Explain the graph given below: **[50 Points]**

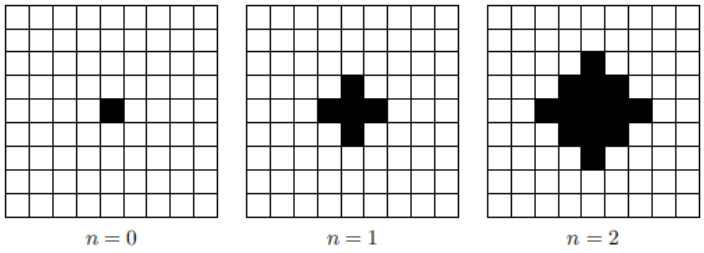


73. What is efficiency of an algorithm dependent on? **[20 Points]**

74. You are facing a wall that stretches infinitely in both directions. There is a door in the wall, but you know neither how far away nor in which direction. You can see the door only when you are right next to it. Design an algorithm that enables

you to reach the door by walking at most O(n) steps where n is the (unknown to you) number of steps between your initial position and the door. **[60 Points]**

75. How many one-by-one squares are generated by the algorithm that starts with a single square and on each of its n iterations adds new squares all round the outside. How many one-by-one squares are generated on the nth iteration? (In the parlance of cellular automata theory, the answer is the number of cells in the von Neumann neighborhood of range n.) The results for n = 0, 1, and 2 are illustrated below: **[70 Points]**



76. Solve: x(n) = x(n/3) + 1 for n > 1, x(1) = 1 (solve for $n = 3^k$) **[30 Points]**

77. Write an algorithm to compute nth Fibonacci number and perform efficiency analysis. **[80 Points]**

78. Write recursive and non-recursive program to compute a^n where a and n are given by the user and perform efficiency analysis. **[50 Points]**

79. In a party of N people, each person will shake her/his hand with each other person only once. On total how many hand-shakes would happen? Explain using efficiency analysis. **[50 Points]**

80. With an example, explain best case, worst case and average case efficiency analysis. **[50 Points]**

Set E – Multiple Choice Questions

[100 Points]

81. Which of the following is not O(n^2)?
a. (15^10) * n + 12099 b. n^1.98 c. n^3 / (sqrt(n)) d. (2^20) * n

82. Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?
f1(n) = 2^n
f2(n) = n^(3/2)
f3(n) = nlogn
f4(n) = n^logn)
a. f3, f2, f1, f4 b. f2, f3, f1, f4 c. f3, f2, f4, f1 d. f2, f3, f4, f1

83. What is the time complexity of the below function?

```
void fun(int n, int arr[]) {
   int i = 0, j = 0;
   for(; i < n; ++i)
      while(j < n && arr[i] < arr[j])
         j++;
}
```

a. O(n^2) b. O(nlogn) c. O(n) d. O(n(logn)^2)

84. In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, same set of statements are executed. Consider the following for loops:
a. for(i = 0; i < n; i++)
b. for(i = 0; i < n; i += 2)
c. for(i = 1; i < n; i *= 2)
d. for(i = n; i > -1; i /= 2)

If **n** is the size of input(positive), which function is most efficient(if the task to be performed is not an issue)?
a. a b. b c. c d. d

85. Consider the following segment of C-code:

```
int j, n;
j = 1;
while (j <= n)
    j = j*2;
```

The number of comparisons made in the execution of the loop for any n > 0 is:
a. CEIL(logn) + 2 b. n c. FLOOR(logn) + 2 d. CEIL(logn)

86. Given two vertices in a graph s and t, which of the two traversals (BFS and DFS) can be used to find if there is path from s to t?
a. Only BFS b. Only DFS
c. Both BFS and DFS d. Neither BFS nor DFS

87. Consider a hash table with 9 slots. The hash function is h(k) = k mod 9. The collisions are resolved by chaining. The following 9 keys are inserted in the order: 5, 28, 19, 15, 20, 33, 12, 17, 10. The maximum, minimum, and average chain lengths in the hash table, respectively, are:
a) 3, 3, and 3 b) 4, 0 and 1 c) 3, 0, 1 d) 3, 0 2

88. Consider a hash table of size 11 that uses open addressing with linear probing. Let h(k) = k mod 11 be the hash function used. A sequence of records with keys 43 36 92 87 11 4 71 13 14 is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. What is the index of the bin into which the last record is inserted?
a) 5 b) 6 c) 7 d) 8

89. Consider a hash function that distributes keys uniformly. The hash table size is 20. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.5.
a) 7 b) 6 c) 10 d) 4

90. Let G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statements is always true?
a) {u,v} must be an edge in G, and u is a descendant of v in T
b) {u,v} must be an edge in G, and v is a descendant of u in T
c) If {u,v} is not an edge in G then u is a leaf in T
d) f {u,v} is not an edge in G then u and v must have the same parent in T

Set F – The Real Challenges

[1000 Points]

91. There are n plants in a garden. Each of these plants has been added with some amount of pesticide. After each day, if any plant has more pesticide than the plant on its left, being weaker than the left one, it dies. You are given the initial values of the pesticide in each plant. Print the number of days after which no plant dies, i.e. the time after which there are no plants with more pesticide content than the plant to their left.

Input Format
The input contains of an integer n. The next line contains n integers, describing the array p where each p denotes the amount of pesticide in plant.

Output Format
Output an integer equal to the number of days after which no plants die.

Sample Input
7
6 5 8 4 7 10 9

Sample Output
2

Explanation
Initially all plants are alive.
Plants = {(6,1), (5,2), (8,3), (4,4), (7,5), (10,6), (9,7)}
Plants[k] = (i,j) => jth plant has pesticide amount = i.

After the 1st day, 4 plants remain as plants 3, 5, and 6 die.
Plants = {(6,1), (5,2), (4,4), (9,7)}

After the 2nd day, 3 plants survive as plant 7 dies.
Plants = {(6,1), (5,2), (4,4)}

After the 3rd day, 3 plants survive and no more plants die.
Plants = {(6,1), (5,2), (4,4)}

After the 2nd day the plants stop dying.

92. The diameter of a tree (sometimes called the width) is the number of nodes on the longest path between two leaves in the tree. It might or not might pass through the root. Write a function to find diameter of the tree.

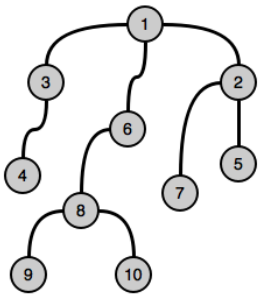
93. Devise an algorithm to detect a cycle in directed graph.

94. A mother vertex in a graph G = (V, E) is a vertex v such that all other vertices in G can be reached by a path from v. Write an algorithm to find the mother vertex.

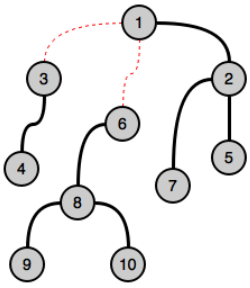
95. Write an algorithm to check if the given binary tree is a binary search tree.

96. You are given a tree (a simple connected graph with no cycles). The tree has N nodes numbered from 1 to N and is rooted at node. Find the maximum number of edges you can remove from the tree to get a forest such that each connected component of the forest contains an even number of nodes.

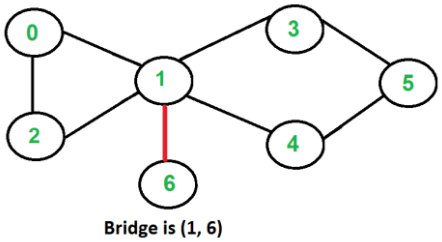
Original tree:



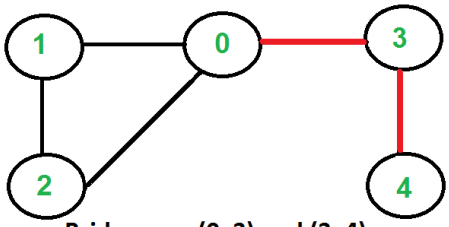
Decomposed tree:



97. An edge in an undirected connected graph is a bridge iff removing it disconnects the graph. For a disconnected undirected graph, definition is similar, a bridge is an edge removing which increases number of connected components. Following are some example graphs with bridges:



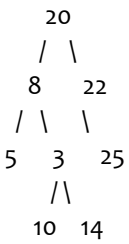
Bridge is (1, 6)



Bridges are (0, 3) and (3, 4)

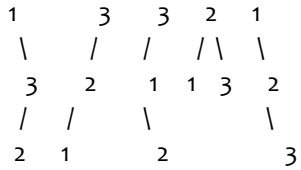
How to find all bridges in a given graph?

98. Given a Binary Tree, we need to print the bottom view from left to right.



The output of above tree is: 5, 10, 3, 14, 25.

99. How many structurally unique BSTs are there for keys from 1..N?
For N = 3, there are 5 possible BSTs. Write an algorithm.



100. Write an algorithm for constructing a Trie.

------ MAY THE FORCE BE WITH YOU ------