Data Structures Job Interview Questions

These are questions that are reported to have been asked during various job interviews. We may not have covered everything that was asked, but you should try to answer them anyway and look up what you do not know.[1]

1. What is a data structure?
2. Suppose you are implementing the address book feature for a cellphone. The address book needs to be kept sorted by person's last name and support fast access when queried by last name. Which of the following data structures would be a good choice to use for storing the address book? Explain why. Which would be the bad choice and why?

   (a) unsorted linked list

   (b) sorted linked list

   (c) binary search tree

   (d) hash table

3. Suppose you are a member of a team of programmers implementing a new text editor. You are in charge of the ”UNDO” feature of the editor. What data structure would you use for storing the list of recent changes made to the document?
4. The linked lists that we discussed had a last node whose next_node link was set to null. In circular linked lists, the next_node link of the last node points to the first node. How would you write C++ code for a method

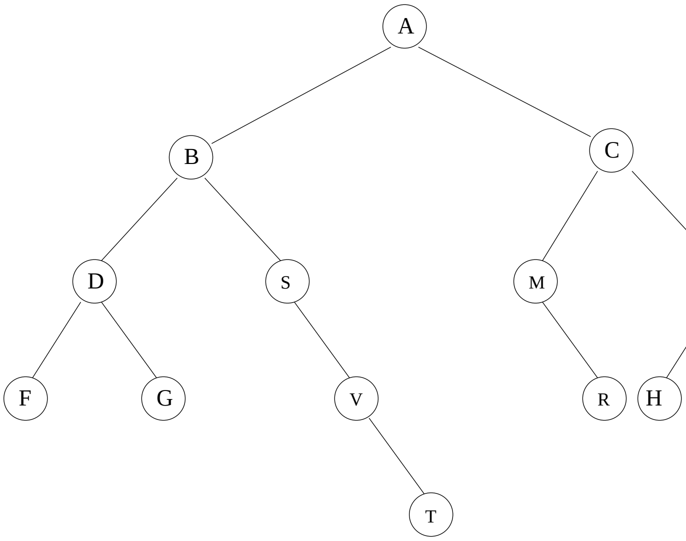
   ```
   boolean isCircular( linkedList)
   ```

   which, given a linked list, returns true if the list is circular and false if it is not?
5. Given a singly linked list, how would you find a node that is `n` nodes away from the last node? Think of iterative and recursive solutions. You can make only a single pass through the list.
6. Given two sorted linked lists, write code that merges them into a single sorted linked list without copying the lists into a new list; i.e., just merge the two into one.
7. How can you simulate queue behavior using two stacks?

[1] Thanks to Joanna Klukowska for collecting many of these questions.

8. Write a method of a binary (search) tree class that returns the sum of all of the numbers stored in the nodes.
   Write another method that returns the sum of the numbers stored in the leaf-nodes.
   Write another method that returns the sum of the numbers stored at even numbered levels (assume that the root is at level 0, which is even).

9. Given the binary tree below show the inorder, preorder and postorder traversals of the tree.



10. Write a method of a binary search tree class that converts the tree to its mirror image (i.e., swaps left and right child for each node). Is the resulting tree a binary search tree?

11. Write a method of a binary tree class that determines if a given tree is a binary search tree or not.

12. Write a method of a binary tree

    ```
    String printAllPathsFromRoot ()
    ```

    that returns a multi-line String containing one per line all paths from root to the leaves (there should be as many lines as there are leaves in the tree).

13. Write a method of a binary (search) tree that returns the largest depth of any node.

14. Given a sorted array (increasing order) of integers, write an algorithm that creates a binary search tree of minimal height.

15. Write an algorithm that when given a link to a node in a binary search tree returns the successor value of that node in the tree (the ”successor” being the value that would follow the value in the given node in an inorder traversal of the tree). Do not use the inorder traversal.

16. Complexity (or Big-O notation)

    (a) What is the complexity (average or worst case) of inserting a new element into:

       i. an unsorted singly linked list (at the end)
       ii. an unsorted doubly linked list (at the end)
       iii. a stack (assume array based)
       iv. a queue (assume array based)
       v. a binary search tree
       vi. a hash table (we did not cover this)

    (b) What is the complexity (average or worst case) of finding a specific element in:

       i. an unsorted singly linked list
       ii. an unsorted doubly linked list
       iii. a sorted singly linked list
       iv. a binary search tree
       v. a hash table (we did not cover this)

17. What is the relationship between a queue and a priority queue? (we did not cover this)

18. What data structure can ”simulate” recursion?

19. How would you design a stack that, in addition to traditional push() and pop() operations, also provides a max() function? Your push(), pop() and max() functions should be O(1).

20. Which sort algorithm guarantees performance better than O(N^2)?

21. Is it possible to implement a binary tree using an array? How?

22. Given a binary search tree how do you find a predecessor/successor of a node?

Additional webpages with resources:

http://www.programcreek.com/2012/11/top-10-algorithms-for-coding-interview/

http://career.guru99.com/top-50-data-structure-interview-questions/

https://www.udemy.com/blog/data-structures-interview-questions/