Joanna Klukowska
joannakl@cs.nyu.edu

Lecture: Data Structures Job Interview Questions

As the semester comes to the end here are some data structures related questions that have been reported as being asked during job interviews. We covered the material that should allow you to answer all of them, but to answer some of these questions you may need to think outside of the proverbial box.

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?

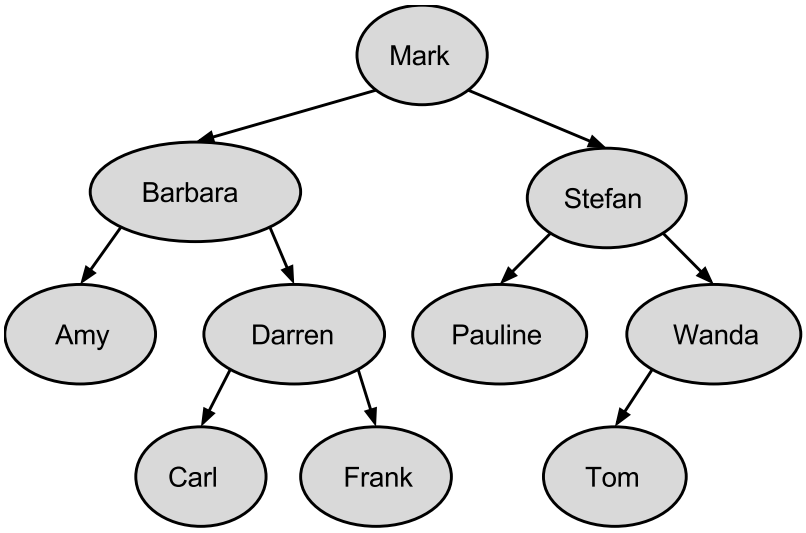
   (a) unsorted linked list
   (b) sorted linked list
   (c) binary search tree
   (d) hash table
3. Suppose you are a member of a team of programmers implementing a new text editor. You are in charge of the "UNDO" feature of the editor. What data structure would you use for storing list of recent changes made to the document?
4. The linked lists that we discussed had a last node whose next reference was equal to null. In circular linked lists, the next reference of the last node points to the first node. How would you write a Java code for method

   ```
   boolean isCircular( linkedList)
   ```

   that 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 (do not copy the lists into a new list, just merge the two into one).
7. How can you simulate queue behavior using two stacks?
8. Write a method of a binary (search) tree class that returns the sum of all 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).

Joanna Klukowska
joannakl@cs.nyu.edu

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 given a reference to a node in a binary search tree returns the ”next” value stored in the tree (”next” defined by which value would come after the value in the given node in the 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. unsorted singly linked list (at the end)
        ii. unsorted doubly linked list (at the end)
        iii. stack (assume array based)

Joanna Klukowska
joannakl@cs.nyu.edu

iv. queue (assume array based)
v. binary search tree
vi. hash table

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

i. unsorted singly linked list
ii. unsorted doubly linked list
iii. sorted singly linked list
iv. binary search tree
v. hash table

17. What is the relationship between a queue and a priority queue?
18. What data structure can ”simulate” recursion?
19. How would you design a stack which, in addition to traditiona 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. Can you represent a binary tree as 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/