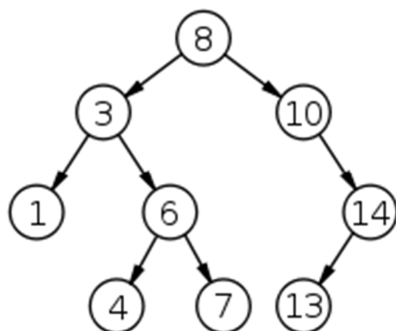


COS 226 – Data Structures and Algorithms
Fall 2014 – Flipped Lecture Section
Individual worksheet
Week 5 – 10.07.14

Instructions: Answer at least 5-10 questions in class using **anonymous** comments. When you find the location of the video that is relevant to answering the question, make a comment, with the title **Question 1** (for example). Write the possible answer to the question (anonymous) and link any other videos that may be relevant to answering the question. You are not expected to answer all questions. Look for answers written by other “anonymous” students and vote for good answers so we can see which answers get the most votes. We will try to compile some of the best answers for Thursdays lecture.

1. State 3 applications where a symbol table data structure can be used.
2. Suppose that a symbol table API (page 363) is implemented with a sorted array. What is the complexity of each operation in symbol table API?
3. Suppose that a symbol table API (page 363) is implemented with a linked list. What is the complexity of each operation in symbol table API?
4. Give a trace of the process of inserting the keys E A S Y Q U E S T I O N into an initially empty table using SequentialSearchST. How many compares are involved?
5. What is a lazy implementation of delete and explain a situation where lazy implementation may not be the best way to do delete in a symbol table?
6. In a symbol table which one is unique? keys or values? why?
7. If a symbol table is implemented with a linked list, what is the complexity of each of the operations listed in the API (as in question 2)?
8. What is the definition of rank(key)?
9. List all operations in an ordered symbol table API.
10. How much memory is required to hold a node of a BST that contains an Item and two pointers?
11. Insert the keys {B, D, A, Z, R, C} in alphabetical order into a BST. Explain all the operations performed (compare, insert, initialize) in the process of building the BST
12. List 4 ordered ways that a BST can be traversed. Show the output when the BST in question #9 is traversed in two of the orders that was described.

13. If keys $\{1,2,3,\dots,n\}$ are inserted into a BST into the order they come in (non-randomized) what is the worst case search complexity?
14. Draw a BST when keys E A S Y Q U E S T I O N is inserted in that order associating value i with i^{th} key.
15. In general BST's cannot handle duplicate keys. Suggest a way to design the data structure so that duplicate keys can be handled. What is the extra cost? memory and/or runtime?
16. If n keys are inserted into a BST in random order what is the expected number of compares to search or insert in tilde notation?
17. Design an algorithm to find the total number of nodes in BST. What is complexity of your algorithm?
18. Design an algorithm to find the height of a BST. What is complexity of your algorithm?
19. Design an algorithm to find the floor and ceiling of a given key. What is the complexity of your algorithm?
20. Given 4 comparable keys $\{1,2,3,4\}$ how many BST's are possible? Draw all the trees.
21. Why is it that the lazy approach of deleting keys from a BST is not desirable? State a situation where the lazy approach can be a problem.
22. Consider the following BST(source: Wikipedia). Show the tree after deleting the keys 1, 14 and 8 successively.



23. Show that it is not possible to build a BST in linear time given n keys
24. What is the worst case complexity of the following operations. State your answer in order of growth notation.
- traverse a BST inorder
 - traverse a BST in level order
 - Find the max of a BST
 - Find the floor of a key
 - Find all keys in a given range $[k_1, k_2]$ including k_1 and k_2
25. Given n horizontal and n vertical line segments, design a naive algorithm for each of the following operations and determine worst case order of growth for your algorithm
- Find all intersection points
 - Find all intersection points between two horizontal or two vertical line segments
26. Explain an efficient algorithm for finding intersection of n horizontal and n vertical line segments. What is the order of growth of your algorithm for finding all intersection points?
27. Given n points (x,y) form, design an algorithm to find all points that are inside a given rectangle? What is the worst case order of growth for your algorithm?
28. Insert the points $(2,3)$, $(2,4)$, $(1,1)$, $(5,2)$, $(3,3)$ into a 2d tree. Show the tree and corresponding bounding box for each point.
29. Describe the operations, nearest neighbor search and range search in 2d trees. What is the worst case order of growth in each case?