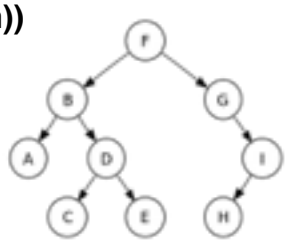
Tuesday, January 10, 2017

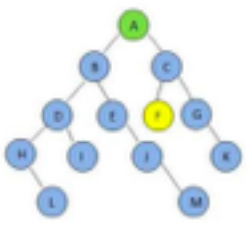
Lecture 2

COMP250 Review Questions

1. Running-time O(m+n) is both O(m) + O(n) and O(max(m, n)) are true
2. Let T1(n) = O(fn)) and T2(n) = O(f(n)), only T1(n)+T2(n)=O(f(n)) is true, T1(n)/T2(n)=O(1) is **not** true
3. sum = 0; for (int i = 0; i < n; i++) { for (j = 1; j < n; j = j*2) { sum += n; } } For the piece of code illustrated, what is the time complexity? **O(n*log(n))**
4. For this binary tree, what is the post-order traversal? **A,C,E,D,B,H,I,G,F**



5. For the Binary Search Tree, deletion of node F would result in which of the following nodes becoming the root node? **E or G**
6. Suppose we need to sort a list of employee records in ascending order, using the social security number (9-digit number) as the key. If we need to guarantee that the running time will be no worse than nlogn, which sorting methods could we use? **Mergesort** because Quicksort can sometimes result in n^2 in worst case scenario
7. Which of the following assertions are true for heaps? **Heaps are binary trees and Heaps are used to implement priority queues.** Heaps are **not** binary search trees and Heaps are **not** used to implement lists.
8. What is the time-complexity of the removal of the highest priority key in a heap (where n is the number of keys stored)? **O(log(n))**
9. You are using a hash table to store keys. Assuming **no collisions**, which of the following operations have a O(1) time complexity? **Insert key, remove key, and find key are all O(1) assuming no collisions**
10. Which of the following assertions are true? **Trees are graphs. A graph that is not a tree has at least one cycle. A Hamiltonian cycle visits each vertex exactly once.** The following assertions are false. **Graphs are trees. A Eulerian cycle visits each vertex exactly once.**
11. In the graph illustrated, starting from the green node at the top, which algorithm will visit the least number of nodes before reaching the yellow node: **Breadth-first search.** Depth-first search would require a lot more visits



12. The following statement submarine.dive(depth), **dive must be a method**

13. T(n) = T(n-1)+O(n) is a recurrence for **Insertion sort**