

Review Questions

Priority Queues / Heaps / Trees / Tree-Traversals

1. A “binary heap” is useful in solving many kinds of problems. For example it is useful for implementing another important data structures: the priority queue. Describe one other problem in which a binary heap is useful.
2. Fill in the blanks: Lists (e.g. Arrays and Linked Lists) are _____ data structures, whereas Trees are _____ data structures.
(Hint: what kind of data structures are these?)
3. Name and briefly describe the two binary heap (assume “min-heap”) constraints.
4. In a binary heap, where are nodes **added**?
5. In a binary heap, which node **replaces** the root node (when it is removed)?
6. By themselves, adding a node to, or removing the root of, a binary heap takes constant time $O(1)$ – because the size of the heap is not taken into account. However, each of these operations have an additional overhead of ensuring the two heap properties are maintained. What is the time complexity of the insertion and removal binary heap operations?
7. What heap property is being taken into consideration when placing a new node on the lowest level, from left to right?
8. What heap property is being taken into consideration when removing an item from the root of the binary heap?

9. Which tree traversal method (on a BST) sorts the values from smallest to largest?
10. Which tree traversal method (on a binary tree) performs depth-first search?
11. Can in-order traversal be performed on a binary tree that is not a binary-search tree?
12. On a BST, how does the insert operation know where to add a new node to the tree?