Questions for recitation 4

Recitation 4 Questions

Try to first do the first 4 questions of recursion and first 6 questions of tree. Do the rest if you have time.

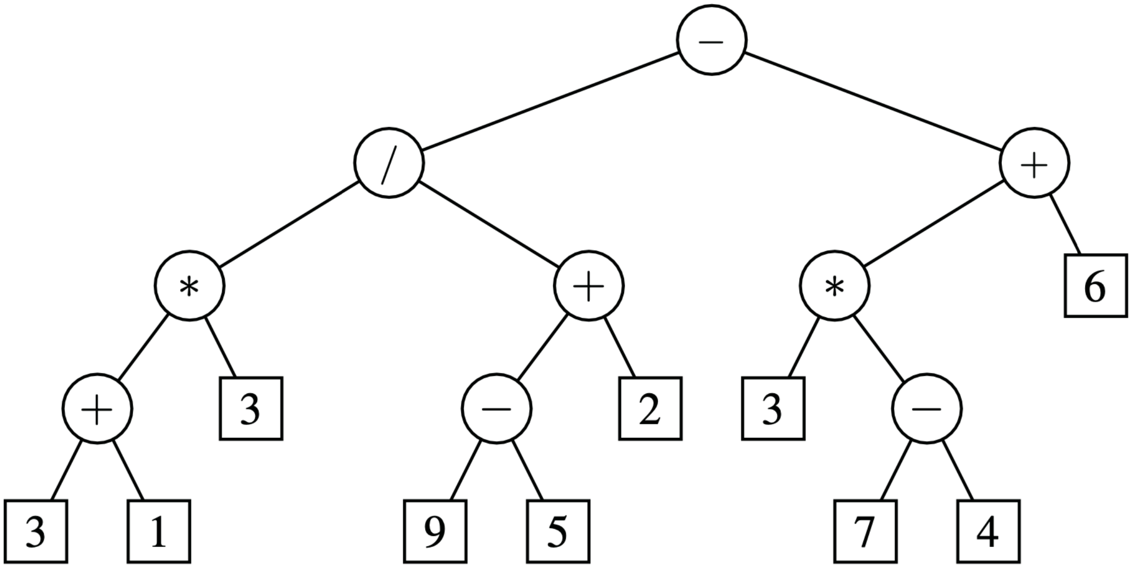
Recursions

1. Describe a recursive algorithm for finding the maximum element in an array, *A*, of *n* elements. What is your running time and space usage?
2. Describe a recursive algorithm to compute the integer part of the base-two logarithm of *n* using only addition and integer division.
3. Write a recursive method that will output all the subsequence of a string of *n* elements (without repeating any subsets). Print the subsequence.
4. Write a short recursive Java method that determines if a string *s* is a palindrome, that is, it is equal to its reverse. Examples of palindromes include 'racecar' and 'gohangasalamiimalasagnahog'.
5. Write a short recursive Java method that rearranges an array of integer values so that all the even values appear before all the odd values.
6. Given an unsorted array, *A*, of integers and an integer *k*, describe a recursive algorithm for rearranging the elements in *A* so that all elements less than or equal to *k* come before any elements larger than *k*. What is the running time of your algorithm on an array of *n* values?
7. Suppose you are given an array, *A*, containing *n* distinct integers that are listed in increasing order. Given a number
   k, describe a recursive algorithm to find two
   integers in
   A that sum to *k*, if such a pair exists. What is the running time of your algorithm?
8. Isabel has an interesting way of summing up the values in an array A of n integers where *n* is a power of two. She creates an array *B* of half the size of *A* and sets $B[i] = A[2i] + A[2i + 1]$, for $i = 0,1,\dots,(n/2) - 1$. If *B* has size 1, then she

outputs *B*[0]. Otherwise, she replaces *A* with *B*, and repeats the process. What is the running time of her algorithm?

Tree

1. Show a tree achieving the worst-case running time for algorithm depth.
2. Describe an algorithm, relying only on the BinaryTree operations, that counts the number of leaves in a binary tree that are the left child of their respective parent.
3. Write the pre-order, in-order, and post-order traversal of the following tree.



4. Draw a binary tree *T* that simultaneously satisfies the following:
   Each internal node of *T* stores a single character.
   • A *preorder* traversal of *T* yields EXAMFUN.
   • An *inorder* traversal of *T* yields MAFXUEN.
5. The path length of a tree T is the sum of the depths of all positions in T. Describe a linear-time method for computing the path length of a tree Tx. (Hint: use recursion with an auxiliary variable)

6. For a tree T , let nI denote the number of its internal nodes, and let nE denote the number of its external nodes. Show that if every internal node in T has exactly 3 children, then $nE = 2nI + 1$.

7. Add support in Linked Binary Tree for a method, swap(p, q), that has the effect of restructuring the tree so that the node referenced by p takes the place of the node referenced by q, and vice versa. Make sure to properly handle the case when the nodes are adjacent.

8. Give an efficient algorithm that computes and prints, for every position p of a tree T , the element of p followed by the height of p's subtree.

9. Design algorithms for the following operations for a binary tree T:
   - preorderNext(p): Return the position visited after p in a preorder traversal of T (or null if p is the last node visited).
   - inorderNext(p): Return the position visited after p in an inorder traversal of T (or null if p is the last node visited).
   - postorderNext(p): Return the position visited after p in a postorder traversal of T (or null if p is the last node visited).

   What are the worst-case running times of your algorithms?

10. Let T be a tree with n positions. Define the ***lowest common ancestor*** (LCA) between two positions p and q as the lowest position in T that has both p and q as descendants (where we allow a position to be a descendant of itself). Given two positions p and q, describe an efficient algorithm for finding the LCA of p and q. What is the running time of your algorithm?

11. The ***indented parenthetic representation*** of a tree T is a variation of the parenthetic representation of T (see Code Fragment 8.26) that uses indentation and line breaks as illustrated in Figure 8.22. Give an algorithm that prints this representation of a tree.