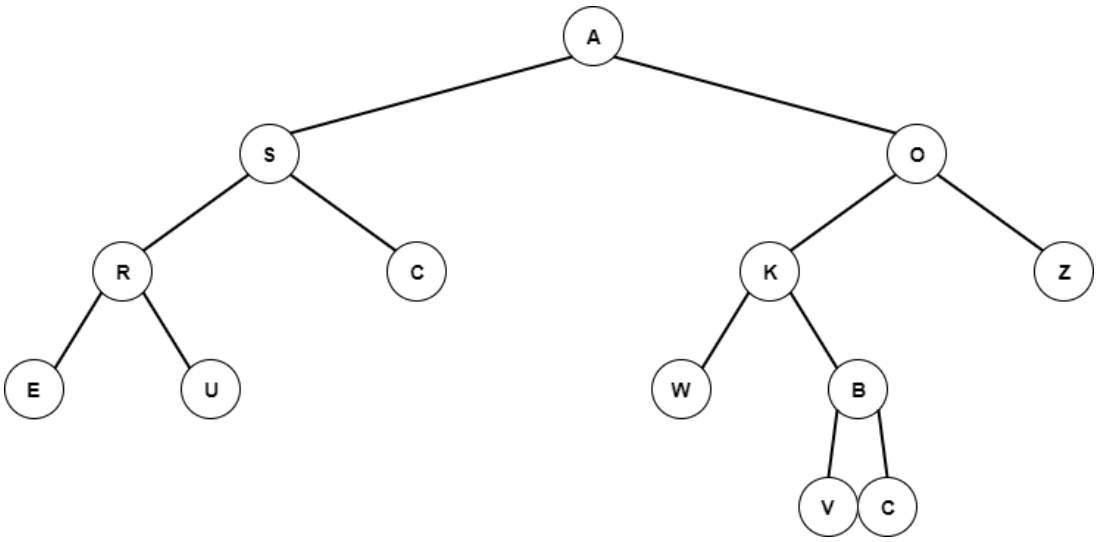
Tutorial (week 6)

All the answers can be given with pseudocode or with Python. Remember that most of the time, many solutions are possible.

1. For the following tree, give
   - the number of nodes
   - the root of the tree
   - the height of the tree
   - the depth of the node with letter W
   - the parent and descendants of the node with letter K
   - the sibling of the node with letter K
   - the external nodes
   - the internal nodes



2. For the previous question's binary tree, give the ordered list of nodes visited
   - during a preorder traversal
   - during an inorder traversal
   - during a postorder traversal
   - during an Euler tour traversal

3. Let T be a proper binary tree such that all the external nodes have the same depth. Let D_e be the sum of the depths of all the external nodes of T, and let D_i be the sum of the depths of all the internal nodes of T. Find constants a and b such that

$$D_e + 1 = aD_i + bn$$

   where n is the number of nodes of T.

4. Design algorithms for the following operations for a node `v` in a proper binary tree `T`:

   - `preorderNext(v)`: return the node visited after `v` in a preorder traversal of `T`
   - `inorderNext(v)`: return the node visited after `v` in an inorder traversal of `T`
   - `postorderNext(v)`: return the node visited after `v` in a postorder traversal of `T`.

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

5. Give an $O(n)$-time algorithm that computes and prints the depth of all the nodes of a tree `T`, where n is the number of nodes of `T`.

6. Describe in Python or pseudocode a nonrecursive method for performing an Euler tour traversal of a proper binary tree that runs in linear time and does not use a stack.

7. Describe in Python or pseudocode a nonrecursive method for performing an inorder traversal of a proper binary tree in linear time.

8. The *path length* of a tree T is the sum of the depths of all the nodes in `T`. Describe a linear-time method for computing the path length of a tree `T` (which is not necessarily binary).

9. Define the *internal path length*, $I(T)$, of a tree `T` to be the sum of the depths of all the internal nodes in `T`. Likewise, define the external path length, $E(T)$, of a tree `T` to be the sum of the depths of all the external nodes in `T`. Show that if `T` is a binary tree with $n(T)$ internal nodes, then $E(T) = I(T)+2n(T)$.

Hints

- **Question 3:** try to gain some intuition by drawing a few different proper binary trees such that all the external nodes have the same depth.
- **Question 4:** think about what could be the worst case number of nodes that would have to be traversed to answer each of these queries.
- **Question 6:** you can tell which visit action to perform at a node by taking note of where you are coming from.
- **Question 7:** use a stack.
- **Question 8:** modify an algorithm for computing the depth of each node so that it computes path lengths at the same time.
- **Question 9:** use the fact that we can build `T` from a single root node via a series of $n(T)$ operations that expand an external node into an internal node with two leaf children.