MECHTRON 2MD3
Data Structures and Algorithms for Mechatronics
Winter 2022

Tutorial 09

Published: Mar 21, 2022

I recommend you try to solve the problems by yourself before the tutorial. Problems will be solved during the tutorial hours. I have uploaded the companion code in the same folder as this document.

1 Question 1

Suppose an array based implementation of the Vector ADT. In the lectures, we have seen that the insert and erase operations take $O(n)$ time which is due to the fact that a significant number of elements must be shifted to perform the operations. But, if we implement the vector in a specific manner it can achieve $O(1)$ time for insertion and removals from index 0 of the vector. Describe the algorithm for the vector ADT that can insert and erase at index 0 of the vector in constant time. Also, provide an algorithm for the $elemAt(j)$ function that returns the j-th element of the vector.

2 Question 2

For each node v in a tree T, let $pre(v)$ be the rank of v in a preorder traversal of T , let $post(v)$ be the rank of v in a postorder traversal of T , let $depth(v)$ be the depth of v, and let $desc(v)$ be the number of descendants of v, not counting v itself. By rank of a node in a specific traversal, we mean the index of a node in the sequence of a traversal. Derive a formula defining $post(v)$ in terms of $desc(v)$, $depth(v)$, and $pre(v)$, for each node v in T.

3 Question 3

One possible question related to the binary tree traversals could be to infer the structure of a binary tree using its inorder traversal and either of preorder or postorder traversals. In this question, we will see how to draw a unique binary tree from the following combination of traversals:

- inorder and preorder
- inorder and postorder

- inorder and level-order (in the next question we will see how level-order traversal works)

The following traversals will be our running examples:

- inorder: 12, 4, 8, 7, 1, 3, 5, 13, 2
- preorder: 8, 4, 12, 5, 1, 7, 3, 13, 2
- postorder: 12, 4, 7, 3, 1, 2, 13, 5, 8

4 Question 4

Design an algorithm to perform the level-order traversal of a binary tree. This traversal is also called a Breadth-First Traversal (BFS), but this name is mostly used in the context of graphs. We call it level-order since it traverses all the nodes at each level before going to the next level (depth).

The level-order traversal of a tree visits nodes level-by-level and at the same level it visits nodes based on their order, if any. In the context of binary trees we know that there is an order between children. Thus, a level-order traversal of a binary tree visits first the left child and then the right child at the same level.

We will see two algorithms to solve this problem, one using recursion and the other using Queue.

5 Question 5

Design an iterative algorithm for the inorder traversal using the Stack data structure.

6 Question 6

In our previous tutorial we have seen how to convert an infix notation of a mathematical expression to a postfix notation. I am sure you are familiar with that algorithm in which we used a stack to solve the problem. In this question, we will design an algorithm to evaluate the resulting postfix notation. The algorithm will again use a stack. I already described the procedure verbally that we have to scan the expression from left to right and stack the operands. Once we see an operator, we should pick two operands from the stack and perform the operator on them and push back the result to the stack.