

National Institute of Technology Calicut
Department of Computer Science and Engineering
Fourth Semester B. Tech.(CSE)-Winter 2023-2024
CS2094D Data Structures Laboratory
Assignment Cycle#1
Part B

Submission deadline (on or before): 01.02.2024, 11:00 PM.

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Ensure that your programs will compile and execute without errors using gcc compiler.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for that evaluation.
- Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding F grade in the course. Detection of ANY malpractice related to the lab course can lead to awarding an F grade in the course.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be

ASSGC<NUMBER>_<PART>_<ROLLNO>_<BATCHNO>_<FIRST-NAME>.zip

(Example: *ASSGC1_A_BxyyyyCS_CS01_LAXMAN.zip*). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

- The source codes must be named as

ASSG<NUMBER>_<PART>_<ROLLNO>_<BATCHNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c

(For example: *ASSGC1_A_BxyyyyCS_CS01_LAXMAN_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: <https://minerva.nitc.ac.in/?q=node/650>.

QUESTIONS

1. Given an inorder and a preorder traversal of a binary tree, construct the unique binary tree T corresponding to the traversals given.

The binary tree contains non-negative integers as data to be sorted.

The program should support the following functions:

- (a) `main()`: Repeatedly reads an input character from the menu list through the terminal and executes menu driven operations accordingly.
- (b) `PostOrder(T)`: Prints the post order traversal of the binary tree T
- (c) `ZIG_ZAG(T)`: Prints the ZIGZAG traversal (Definition is given below) of the binary tree T
- (d) `Level_Max(T)`: Prints maximum element at each level of the binary tree T.
- (e) `Diameter(T)`: Prints the diameter of the binary tree T
- (f) `Left_Leaf_Sum(T)`: Prints the sum of the values of left leaf nodes.

Definitions:

Depth of a node: Length(no of edges) of the unique simple path from the root to the node

Level of a tree: All the nodes at the same depth form a level of the tree. Root is at level 0 and the nodes at depth 1 are at level 1, nodes at depth 2 are at level 2.

ZIG-ZAG Traversal of a Binary Tree: It is a level order traversal in which if the level no is even, nodes will be printed from left to right and if the level no is odd, nodes will be printed from right to left. Please note that level no starts from 0.

DIAMETER of tree :The diameter/width of a tree is defined as the number of nodes on the longest simple path between two end nodes.

Left Leaf Node: A leaf node which is present as a left child to its parent node.

Input Format:

- The first line is an integer $n \in [1, 10^6]$ indicating the number of nodes in the tree.
- Second line of the input contains a space separated integer sequence of “n” numbers representing INORDER traversal of the tree T with key values $\in [1, 10^6]$
- Third line of the input contains a space separated integer sequence of “n” numbers representing PREORDER traversal of the tree T with key values $\in [1, 10^6]$
- Next Each line contains a character [**p**’,**z**’,**m**’,**d**’,**s**’, **e**’].
 - Character ‘p’ call - `Postorder(T)` - is to print the post order traversal of a tree.
 - Character ‘z’ call - `ZIG_ZAG(T)` - is to print the zig-zag traversal of the tree.
 - Character ‘m’ call - `Level_Max(T)` - is to print max elements at each level.
 - Character ‘d’ call - `Diameter(T)` - is to print the diameter of a binary tree.
 - Character ‘s’ call - `Left_Leaf_Sum(T)` - is to print the sum of all left leaf nodes.
 - Input ‘e’ terminates the execution of the program.

Output Format:

- Lines contains appropriate output sequence of integer or an integer output

Sample Input 1:

```
7
4 2 5 1 6 3 7
1 2 4 5 3 6 7
p
z
m
d
s
e
```

Sample Output 1:

```
4 5 2 6 7 3 1
1 3 2 4 5 6 7
1 3 7
5
10
```

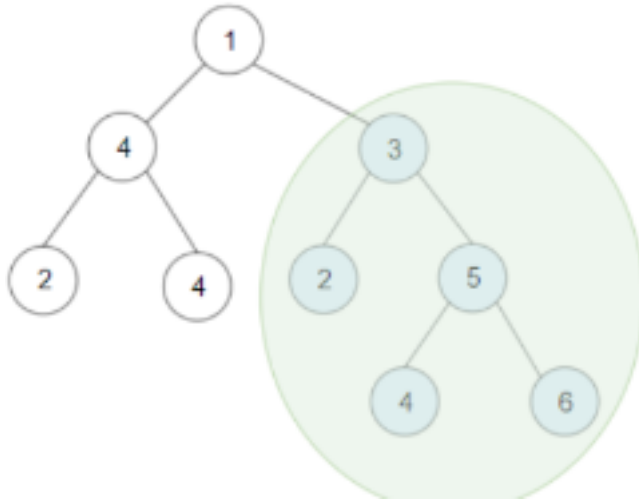
Sample Input 2:

```
4
4 2 5 1
1 2 4 5
z
p
s
d
e
```

Sample Output 2:

```
1 2 4 5
4 5 2 1
4
3
```

2. Given the PARENTHESIS REPRESENTATION of the binary Tree T with n nodes (where $n \in [1, 10^4]$). Write a menu driven program to implement the operations outlined below:
- (a) Levelordertraversal(T): This function prints the level order traversal of the binary tree. Level order traversal of a tree is breadth-first traversal for the tree.
 - (b) MaximumSumBst(T): This function computes the maximum sum of all keys of any sub-tree which is also a Binary Search Tree (BST).
 - (c) RightView(T): This function prints the Right view of a Binary Tree which is a set of nodes visible when the tree is viewed from the right side.



Level order Traversal : 1 4 3 2 4 2 5 4 6

Maximum Sum of BST: 20

Right view : 1 3 5 6

Input Format:

- First line of the input contains Parenthesis Representation of the tree T with key values $\in [1, 10^6]$.
- Next Each line contains a character 'l', 'm', 'r' or 'e'.
- Character 'l' is to print the level order traversal of Tree
- Character 'm' is to print the maximum sum of BST in a binary tree.
- Character 'r' is to print the right view of the binary tree.
- Input 'e' terminates the execution of the program.

Output Format:

- The output (if any) of each command should be printed on a separate line.

Sample Input 1:

```

1(4(2)(4))(3(2)(5(4)(6)))
l
r
m
e

```

Sample Output 1:

```

1 4 3 2 4 2 5 4 6
1 3 5 6
20

```

Sample Input 2:

```

4(2(1)(3))(6(5)(7))
l
m
r
e

```

Sample Output 2:

```

4 2 6 1 3 5 7
28

```

3. Given the PARENTHESIS REPRESENTATION of a binary search tree(BST) T with n nodes (where $n \in [1, 10^3]$). Write a menu driven program to implement the operations outlined below:

- (a) PrintRangeValues(T, k1,k2):This function takes two key values k1 and k2 ($k1 \neq k2$) that specify the range of key values as input arguments and prints the key values of the nodes that are in the given range(inclusive of k1 and k2).
- (b) Ancestors(T,k): This function takes key value k as an input argument and prints all the ancestors of k in T where $k \in [1, 10^6]$ (order starting from the node itself up to the root).

Ancestor of a node x: We call any node y on the unique simple path from the root of the binary tree to x an ancestor of x. Please note that x is also an ancestor of itself.

Input Format:

- First line of the input contains Parenthesis Representation of the tree T with key values $\in [1, 10^6]$.
- Next each line contains a character 'p' (followed by an integer), 'a' (followed by two integers) or 'e'.
- Character 'p' followed by two integer values (key values) is to print the key values of the nodes that are in the given range.
- Character 'a' followed by one integer value(key value) is to print all the ancestors of k in BST.
- Input 'e' terminates the execution of the program.

Output Format:

- The output (if any) of each command should be printed on a separate line.

Sample Input 1:

```
7(5(2)(6))(9(8)(10))
p 5 8
a 8
e
```

Sample Output 1:

```
5 6 7 8
8 9 7
```

Sample Input 2:

```
8(5()(6))(10()(11))
a 6
p 8 11
e
```

Sample Output 2:

```
6 5 8
8 10 11
```