Homework lecture 4

Trees

1. Given a tree, your task is to write a program with following functions
   - Calculate the height of the tree.

   - Write out the order of nodes following the preorder traversal.

   - Write out the order of nodes following the postorder traversal.

   - Check if the given tree is a binary tree? If yes, write out the order of nodes in inorder traversal.

Input: Data come from the keyboard:

- The first line contains integer numbers N, M indicating the number of nodes, and edges, respectively.

- M following lines each contains two integer numbers u, v indicating that u is the parent of v.

Output: Data are written to the screen as following:

- The height of the tree

- The order of nodes in preorder traversal

- The order of nodes in postorder traversal

- The order of nodes in the inorder if the tree is binary. Otherwise, write the string 'NOT BINARY TREE'

Keyboard	Screen
5 4	2
1 2	1 2 4 5 3
1 3	4 5 2 3 1
2 4	4 2 5 1 3
2 5	

2. Given a list of integer numbers: 2, 1, 10, 6, 3, 8, 7, 13, 20.

- Draw the binary search tree

- Draw the binary search tree after inserting values: 14, 0, 35

- Draw the binary search tree after deleting: 6, 13, 35

3. Given a list of integer numbers: 2, 1, 10, 6, 3, 8, 7, 13, 20.

- Draw the heap tree

- Draw the heap tree after inserting values: 14, 0, 35

- Draw the heap tree after deleting: 6, 13, 35

4. Use random.org to generate a set of 10 integers from 1-20 (S1).

- Insert elements from S1 to a binary search tree one by one and draw the binary search tree after each step.

- Write out the procedure to find and remove the maximum element from binary search tree **in detail.**

- Write out the procedure to find and remove the minimum element from binary search tree **in detail.**

5. Use random.org to generate another set of 10 integers from 1-20 (S2).

- Draw the heap (tree) from S2

- Insert elements from S1 to this heap one by one and draw the heap after each step.

- Write out the procedure to find and remove the maximum element from binary search tree **in detail.**