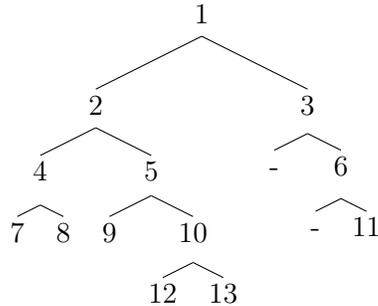


Bilkent University - CS202 - Spring 2013

Quiz 3 - Section 1 - Answer Key

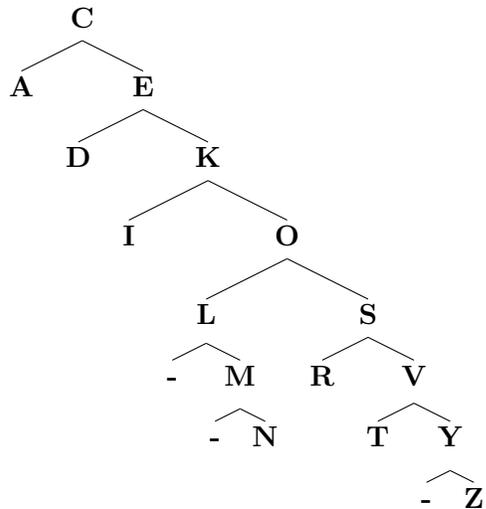
1. (20 points) Given a tree (ignore dashes, i.e., '-' symbols):



Answer the following questions using the tree above. Incorrect answers for yes/no questions (except the first one) will be penalized with -2 points.

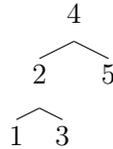
- (1 point) - Did you like this tree (yes/no)? **Yes or No.**
- (2 points) - Is it a binary tree (yes/no)? **Yes.**
- (2 points) - Is it a binary search tree(BST) (yes/no)? **NO.**
- (5 points) - What is the preorder of the nodes? **1, 2, 4, 7, 8, 5, 9, 10, 12, 13, 3, 6, 11.**
- (5 points) - What is the inorder of the nodes? **7, 4, 8, 2, 9, 5, 12, 10, 13, 1, 3, 6, 11.**
- (5 points) - What is the postorder of the nodes? **7, 8, 4, 9, 12, 13, 10, 5, 2, 11, 6, 3, 1.**

2. (40 points) Consider the string: **CEKOSLOVAKYALILASTIRAMADIKLARIMIZ-DANMISINIZ**. You should construct a BST using this string. Each node of the BST will contain a capital letter (not a number). The key order of BST will be the alphabetical order of the letters. Construct the tree by adding letters of the string one-by-one. **At any point, if any letter already exists in the BST, then skip it (no need for readdition).**



Ignore dashes, i.e., '-' symbols.

3. (40 points) Given a binary search tree:



This BST was created by incrementally adding the numbers in the following order: $\{4, 2, 1, 3, 5\}$. Find all possible permutations of the numbers $\{1, 2, \dots, 5\}$ which will create the BST above.

In total, there are 8 permutations that create the BST above:

- 4 5 2 1 3
- 4 5 2 3 1
- 4 2 1 3 5
- 4 2 1 5 3
- 4 2 3 1 5
- 4 2 3 5 1
- 4 2 5 1 3
- 4 2 5 3 1

You could observe that 4 must be inserted before all other numbers. Moreover, 2 must be inserted before 1 and 3. All permutations that satisfy these rules are given above.

Bonus 1. (5 points) Draw an example tree (anything you'd like) that contains 5 nodes and 5 edges. **Theorem states that a tree with m nodes has at most $m - 1$ edges. Otherwise, there will be a cycle. Therefore, it is impossible to draw a tree with 5 nodes and 5 edges.**

Bonus 2. (15 points) Implement a recursive function

```
int width(TreeNode *root, int k);
```

that will take a pointer to the root of the binary tree, an integer k as parameters and will return the width of the tree at level k (number of nodes at level k). Level of the root node is assumed to be level 0. No helper function is allowed. Tree structure:

```
struct TreeNode {
    int data;
    TreeNode *left;
    TreeNode *right;
};
```

```
int width(TreeNode *root, int k) {
    if (root == NULL)
        return 0;

    if (k == 0)
        return 1;

    return width(root->left, k - 1) + width(root->right, k - 1);
}
```