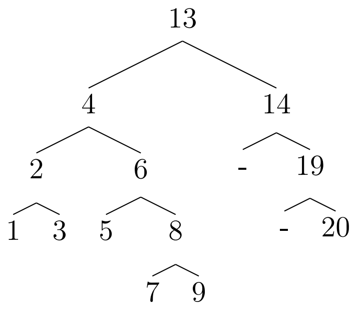
Bilkent University - CS202 - Spring 2013
Quiz 3 - Section 2 - Answer Key

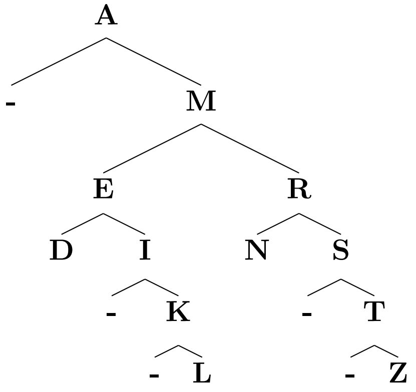
1. (20 points) Given a tree (ignore dashes, i.e., '-' symbols):



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

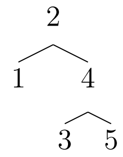
- (1 point) - Did you like this tree (yes/no)? **Yes or No.**
- (2 points) - Is it a binary tree (yes/no)? **Yes.**
- (2 points) - Is it a binary search tree(BST) (yes/no)? **Yes.**
- (5 points) - What is the preorder of the nodes? **13, 4, 2, 1, 3, 6, 5, 8, 7, 9, 14, 19, 20.**
- (5 points) - What is the inorder of the nodes? **1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 14, 19, 20.**
- (5 points) - What is the postorder of the nodes? **1, 3, 2, 5, 7, 9, 8, 6, 4, 20, 19, 14, 13.**

2. (40 points) Consider the string: **AMERIKALILASTIRAMADIKLARIMIZDANMISINIZ**. 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: $\{2, 4, 1, 3, 5\}$. Find all possible permutations of the numbers $\{1, 2, ..., 5\}$ which will create the BST above.

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

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

You could observe that 2 must be inserted before all other numbers. Moreover, 4 must be inserted before 3 and 5. All permutations that satisfy these rules are given above.

Bonus 1. (5 points) Forest is a set of trees. We have a *forest* with t trees. Total number of nodes in the forest is n. Write the number of edges in the forest in terms of t and n. Show your work.
Let's denote the number of nodes in i^{th} $(1 \leq i \leq t)$ tree of the forest as a_i.
Theorem states that a tree with m nodes has $m - 1$ edges.
Then, i^{th} $(1 \leq i \leq t)$ tree of the forest has $a_i - 1$ edges.
Therefore, the total number of the edges in the forest is

$$\sum_{i=1}^{t}(a_i - 1) = \sum_{i=1}^{t} a_i - \sum_{i=1}^{t} 1 = n - t.$$

Bonus 2. (15 points) Implement a recursive function

void merge(TreeNode *root1, TreeNode *root2);
that will take 2 pointers to BST's and will merge the second BST into first BST without allocating any new memory/node. (**Hint:** change the pointers of already allocated nodes.) No helper function is allowed. Tree structure:

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

```
void merge(TreeNode *root1, TreeNode *root2) {
        if (root1 == NULL) {
            root1 = root2;
            return;
        }
        if (root2 == NULL)
            return;

        TreeNode *root2Left = root2→left;
        TreeNode *root2Right = root2→right;
        root2→left = root2→right = NULL;

        TreeNode *temp = root1;
        while (true) {
            if (temp→data > root2→data) {
                if(temp→left == NULL) {
                    temp→left = root2;
                    break;
                } else {
                    temp =temp→left;
                }
            } else if (temp→data < root2→data) {
                if(temp→right == NULL) {
                    temp→right = root2;
                    break;
                } else {
                    temp =temp→right;
                }
            } else {
                break; //skip duplicate nodes
            }
        }
        merge(root1, root2Left);
        merge(root1, root2Right);
}
```