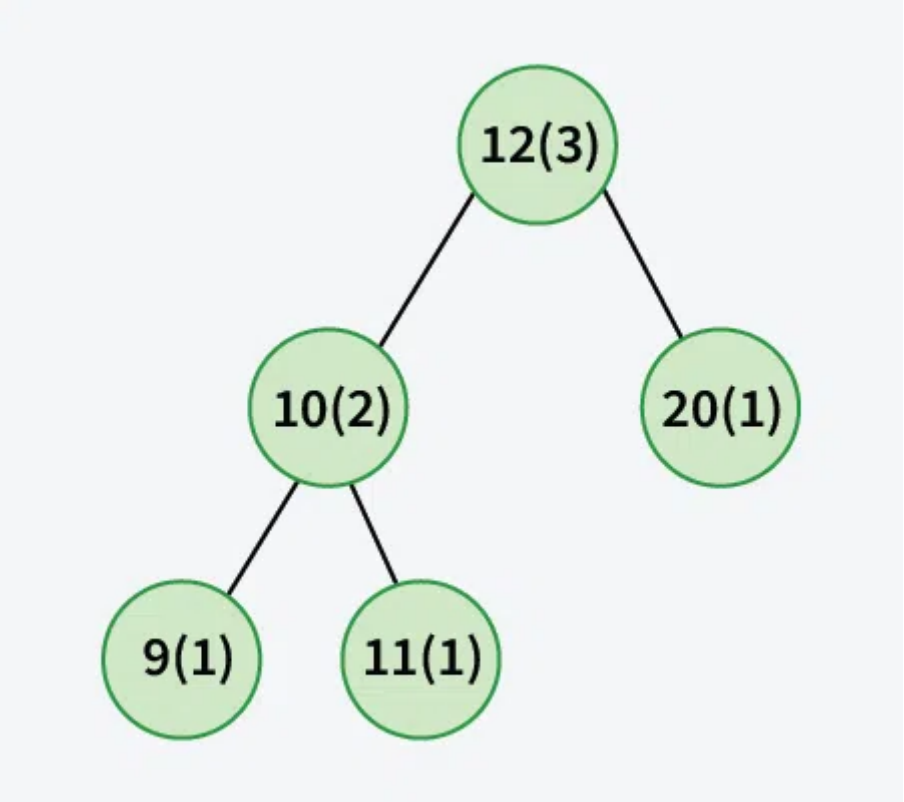
Duplicate Binary Search Tree

Author: Junathan Richie

Time limit: 1 s



Buatlah sebuah Binary Search Tree (BST) yang mampu menyimpan elemen duplikat. Apabila sebuah elemen yang sudah ada dimasukkan ke dalam sebuah pohon, pohon akan menambah jumlah kemunculan dari node yang sudah ada.

BST yang dibuat harus mendukung operasi berikut:

1. insert X
   Menambahkan elemen X ke dalam BST, jika elemen sudah ada, tambahkan jumlahnya sebanyak 1
2. remove X
   Menghapus elemen X dari BST. Jika jumlah elemen lebih dari 1, kurangi jumlahnya
3. inorder
   Mencetak secara transversal inorder dengan format data(jumlah)
4. preorder
   Mencetak secara transversal preorder dengan format data(jumlah)
5. postorder
   Mencetak secara transversal postorder dengan format data(jumlah)

Format Masukan

Baris pertama berisi sebuah bilangan bulat N yang merupakan jumlah perintah yang akan dijalankan

N baris berikutnya merupakan perintah seperti yang dijelaskan di atas

Format Keluaran
Untuk transversal, cetak urutan node sesuai dengan format data(jumlah), dipisahkan dengan spasi

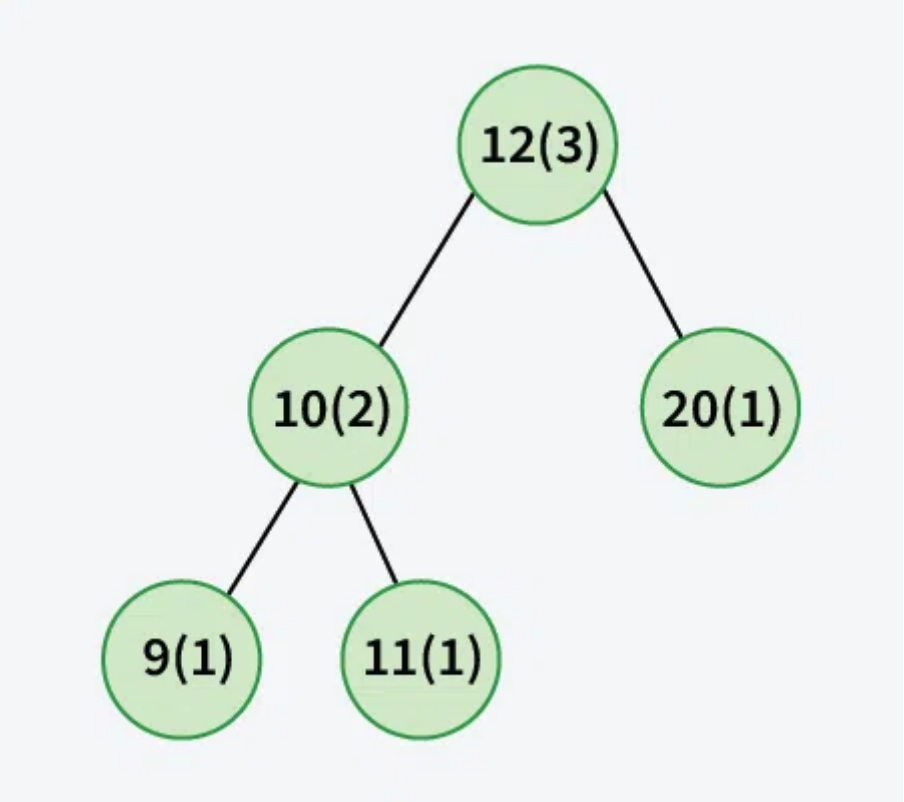
Constraint
$1 \leq n \leq 100$
$-10^9 < X < 10^9$

Sample Input 0
16 insert 12 insert 12 insert 12 insert 10 insert 10 insert 9 insert 11 insert 20 inorder preorder postorder remove 9 remove 10 inorder preorder postorder
Sample Output 0
9(1) 10(2) 11(1) 12(3) 20(1) 12(3) 10(2) 9(1) 11(1) 20(1) 9(1) 11(1) 10(2) 20(1) 12(3) 10(1) 11(1) 12(3) 20(1) 12(3) 10(1) 11(1) 20(1) 11(1) 10(1) 20(1) 12(3)

Duplicate Binary Search Tree

Author: Junathan Richie

Time limit: 1 s



Create a Binary Search Tree (BST) that can store duplicate elements. If an element that already exists is inserted into the tree, the tree should increase the occurrence count of the existing node rather than adding a new one.

The BST must support the following operations:

1. insert X
   Inserts element X into the BST. If the element already exists, increment its count by 1.
2. remove X
   Removes element X from the BST. If the element's count is greater than 1, decrease the count
3. inorder
   Prints an inorder traversal with each node in the format value(count)
4. preorder
   Prints a preorder traversal with each node in the format value(count)
5. postorder
   Prints a postorder traversal with each node in the format value(count)

Format Masukan

The first line contains an integer N, the number of commands to be executed.

The next N lines contain one command per line, as described above.

Format Keluaran

For each traversal command, print the nodes in the specified order using the format value(count), separated by spaces.

Constraint

$1 \leq n \leq 100$

$-10^9 < X < 10^9$

Sample Input 0
16 insert 12 insert 12 insert 12 insert 10 insert 10 insert 9 insert 11 insert 20 inorder preorder postorder remove 9 remove 10 inorder preorder postorder
Sample Output 0
9(1) 10(2) 11(1) 12(3) 20(1) 12(3) 10(2) 9(1) 11(1) 20(1) 9(1) 11(1) 10(2) 20(1) 12(3) 10(1) 11(1) 12(3) 20(1) 12(3) 10(1) 11(1) 20(1) 11(1) 10(1) 20(1) 12(3)