Modification Question II
AVL Tree Operations

Problem

In a secure digital vault system, numbers are stored in such a way that the vault always stays balanced for quick access and retrieval ($O(logn)time$). The vault automatically reorganizes itself every time a unique new number ($n \in [1, 10^6]$) is added to ensure that no section of the vault is overcrowded, making it easier to manage and access the numbers.As a system administrator, your tasks include:

(a) `Adding Numbers to the Vault (insert(V,n))`: Each time you add a number n, the vault V might automatically rearrange itself to maintain the balance.

(b) `PostOrder(V)`: Prints the post order traversal of the Vault V.

(c) `Analyzing Vault Levels (odd(V,k))`: You are occasionally required to analyze specific levels (root of vault at level 0) within the vault. For a given level k, you need to find the average value of all the odd numbers stored at that level in the vault V. If the level does not exist, you should report -1.0. If there are no odd numbers at the requested level, the average should be considered as 0.0.

Input Format:

1. Each line contains a character from {'i','p','a','e'} followed by zero or one integer n, where $n \in [1, 10^6]$.
2. Character 'i' is followed by a positive integer n separated by a space. Perform `insert(V, n)` operation.
3. Character 'p' prints the postorder traversal of the vault.
4. Character 'a' calls `odd(V,k)` -print the average of odd values at the given level or -1.0 or 0.0 accordingly.
5. Character 'e' terminates the execution of the program.

Output Format

The output of each command should be printed on a separate line.However no output is printed for 'i' and 'e'.

- For option 'p', print the postorder traversal of V. Each node's value is separated by a space.
- For option 'a', print the average.

Testcase

Input1:

i 8
i 6
i 4
i 12
i 2
i 13
p
a 2
e

Output:

2 4 8 13 12 6
13.0

Input2:

i 10
i 20
i 40
i 60
i 75
i 80
i 50
i 5
p
a 2
e

Output:

5 10 40 50 60 80 75 20
5.0