**CSE225L – Data Structures and Algorithms Lab**

**Final**

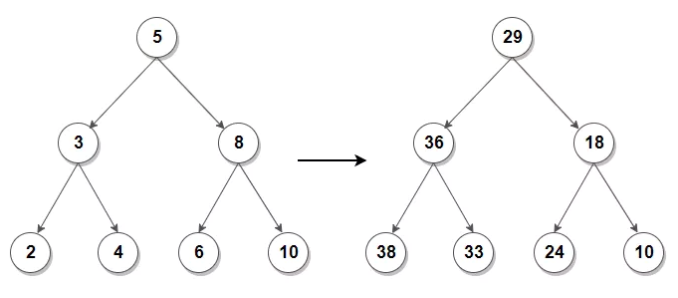
**Name:**

**ID:**

1. Update every key in BST to contain sum of all greater keys

Given a binary search tree, modify it such that every key is updated to contain sum of all greater keys present in BST.

For example, BST shown on the left should be updated to BST on the right.



HINT: We can solve this problem by in-order traversal by calculating the sum of all nodes present in a BST in advance. Then for each node, sum of all greater keys for any node can be updated in constant time using total sum and sum of nodes visited so far.

1. Jesse loves cookies. He wants the sweetness of all his cookies to be greater than value **K**. To do this, Jesse repeatedly mixes two cookies with the least sweetness. He creates a special combined cookie with:

*Sweetness =* (1\* Least *sweet cookie* + 2\**2nd least sweet cookie*).

He repeats this procedure until all the cookies in his collection have a sweetness ≥ **K**  
you are given Jesse's cookies. Print the number of operations required to give the cookies a sweetness ≥ **K**. Print -1 if this isn't possible.

**Input Format**

The first line consists of integers **N**, the number of cookies and **K**, and the minimum required sweetness, separated by a space.   
The next line contains **N** integers describing the array **A** where **Ai** is the sweetness of the  **ith** cookie in Jesse's collection.

**Output Format**

Output the number of operations that are needed to increase the cookie's sweetness ≥ **K**  
Output  -1 if this isn't possible.

**Sample Input**

6 7

1 2 3 9 10 12

**Sample Output**

2