

Cairo University
Faculty of Engineering
Computer Engineering
Fall 2019

Design and Analysis of Algorithms
Homework 3

DEADLINE Saturday, November 16th 2019, 11:59 PM

Your code should be taking into consideration the following requirements:

1. Implement your code in C++.
2. The grade will be granted based on the running time of your code as well as the correct output.
3. Remember! Plagiarism is not tolerated. Any sign of cheating or plagiarism will be graded as ZERO in this assignment and all other assignments.

Deliverables:

- You will submit your code on SPOJ in the contest sent by email.

Requirements:

1. Given an array with duplicates, output the number of ordered pairs where each number's frequency is at least the value of the other. Use hash tables to implement your solution.

The input format will be as follows:

- One line containing the elements of the array

You can assume that the array size will vary between 1 and 10^3 .

Output format should be:

- The count of pairs in on line

Sample Input:

1 1 2 2 3

Sample Output:

4

Output explanation:

Here the pairs are:

- (1, 1): 1 has a frequency of 2 which is bigger than 1.
- (1, 2): 1 has a frequency of 2 which is equal to 2, and 2 has a frequency of 2 which is bigger than 1.
- (2, 1): Similar to (1,2)
- (2, 2): 2 has a frequency of 2 which is equal to 2.

2. Given N numbers and a ratio R , output the count Geometric Progression (Geometric Sequence) subsequences of size 3 and ratio R . Use hash tables to implement your solution.

The input format will be as follows:

- One line containing the ratio R
- One line containing the elements of the array

You can assume that the array size will vary between 3 and 10^7 .

Output format should be:

- The count of subsequences of size 3 and ratio R in one line

Sample Input:

2

1 1 2 2 4

Sample Output:

4

Output explanation:

- Here the only subsequence of size 3 and ratio 2 is (1,2,4). Any of the 1s can be chosen, and any of the 2s can be chosen and only 4 can be the last element. So the count is $2*2*1 = 4$.

Another Example:

Input:

2

1 1 2 2 4 8

Output:

6

Output explanation:

- Here the subsequences of size 3 and ratio 2 are (1,2,4) and (2,4,8). There are 4 ways to generate (1,2,4) and there is 2 ways to generate 2,4,8. The output is 6 (2+4).

3. Given a binary tree calculate the average value of the nodes in each level of the tree. You should store the tree in your code as a tree data structure.

The input format will be as follows:

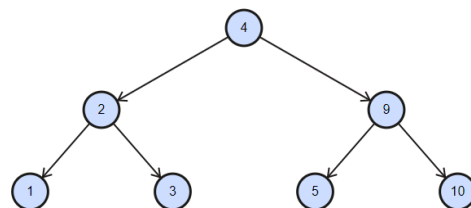
- The first line will contain the number of nodes in the tree N .
- The following N numbers will contain the data in each node.
- The next line will contain the number of edges in the tree E .
- The following E lines will contain 3 values:
 - L or R to represent if this is a left or right child
 - Index of the parent.
 - Index of the child.
- Assume node 0 is the root.

You can assume that the number of nodes will vary between 1 and 10^7 .

Output format should be:

- L (number of levels) numbers, each number represents the average of each level.

Example:



Sample Input:

```
7
4 2 9 1 3 5 10
6
L 0 1
R 0 2
L 1 3
R 1 4
L 2 5
R 2 6
```

Sample Output:

```
4 5.5 4.75
```

4. Given a BST return the in-order traversal of the corresponding greater sum tree. You should store the tree in your code as a tree data structure. Greater sum tree is a tree in which every node contains the sum of all the nodes which are greater than the node in the original BST.

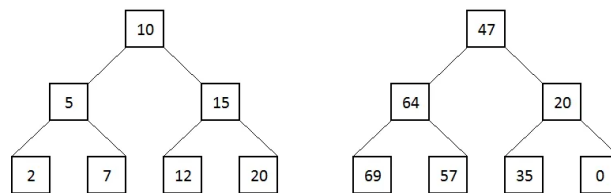
Input Format will be as follows:

- The first line will contain the number of nodes in the tree N.
- The following N lines will contain the data in each node following the in-order traversal of the tree.
- The next line will contain the number of edges in the tree E.
- The following E lines will contain 3 values:
 - L or R to represent of this is a left or right child
 - Index of the parent.
 - Index of the child.
- Assume node 0 is the root.

You can assume that the number of nodes will vary between 1 and 10^7 .

Output format should be N numbers corresponding to the in-order traversal of the greater sum tree.

Example:



Sample Input:

```
7
10 5 15 2 7 12 20
6
L 0 1
R 0 2
L 1 3
R 1 4
L 2 5
R 2 6
```

Sample Output:

```
69 64 57 47 35 20 0
```