BST maintenance

by zxqfd555

Consider a binary search tree *T* which is initially empty. Also, consider the first `N` positive integers {1, 2, 3, 4, 5, ..., N} and its permutation P {a_1, a_2, ..., a_N}.

If we start adding these numbers to the binary search tree *T*, starting from a_1, continuing with a_2, ... (and so on) ..., ending with a_N. After every addition we ask you to output the sum of distances between every pair of *T*'s nodes.

Input Format
The first line of the input consists of the single integer **N**, the size of the list.
The second line of the input contains **N** single space separated numbers the permutation a_1, a_2, ..., a_N itself.

Constraints
$1 \le N \le 250000$

Output Format
Output *N* lines.
On the i^{th} line output the sum of distances between every pair of nodes after adding the first i numbers from the permutation to the binary search tree *T*

Sample Input #00

```
8
4 7 3 1 8 2 6 5
```

Sample Output #00

```
0
1
4
10
20
35
52
76
```

Explanation #00

After adding the first element, the distance is `0` as there is only 1 element

```
4
```

After adding the second element, the distance between 2 nodes is `1`.

```
4
 \
  7
```

After adding the third element, the distance between every pair of elements is `2+1+1=4`

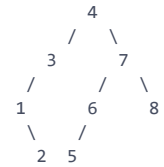
After adding the fourth element, the distance between every pair of elements is `3 + 2 + 1 + 2 + 1 + 1 = 10`

After adding the fifth element, the distance between every pair of elements is `4 + 3 + 2 + 1 + 3 + 2 + 1 + 2 + 1 + 1 = 20`

After adding the sixth element, the distance between every pair of elements is `5 + 4 + 3 + 2 + 1 + 4 + 3 + 2 + 1 + 3 + 2 + 1 + 2 + 1 + 1 = 35`

After adding the seventh element, the distance between every pair of elements is `5+5+4+3+2+1+4+4+3+2+1+3+3+2+1+2+2+1+1+1+2=52`

After adding the final element, the distance between every pair of elements is `6+5+5+4+3+2+1+5+4+4+3+2+1+4+3+3+2+1+3+2+2+1+2+1+1+2+1+3=76`



Submissions:430

Max Score:140

Difficulty: Advanced

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 7

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  public class Solution {
8
9      public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code