**PROBLEMA ADICIONAL PARA PUNTOS EXTRAS**

Como lo comentamos, hace algunos años me tocó diseñar este problema para el concurso regional de programación de la ACM. Al ser un caso interesante de representación de un árbol binario, he querido compartirlo para que intenten resolverlo.

Quien lo desee, puede usar el juez en línea para que se lo revise a través de este link: [https://icpcarchive.ecs.baylor.edu/index.php?option=com\_onlinejudge&Itemid=8&page=show\_problem&problem=273](https://icpcarchive.ecs.baylor.edu/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=273" \t "_blank)

No caigas en la tentación de resolver el problema con ayuda. La idea de este ejercicio es que sea un reto que resuelvas en forma personal.

2001 ACM Programming Contest

Mexico and Central America Region

BST in a matrix

Files: bst.in, bst.out, bst.c, bst.cpp, ..

**Background**

The structure (or shape) of a Binary Search Tree (BST) can be represented in memory as follows:

The keys of the BST are d1, d2, d3, … dn, where d1 < d2 < d3 < … dn .

The matrix M, named the “matrix of roots” for the BST contains the following values: for every i <=j, M[i,j] contains the index of the key that is the root in the BST that is formed with the keys form di to dj. The rest of the values in the matrix M are zero, because you can’t construct a BST with that conditions.

The matrix of roots has rows from *1* to *n+1* and columns from *0* to *n*, to facilitate the construction of the BST if you needed.

The next, is an example of a BST and its matrix of roots:





# Input

You will be given a number of cases. The input for each case consists of one line with the order of the matrix of roots that represents the BST, following with the values of the matrix, one line per row, and each row consist of the values of each column (separated by a tab). All values are non-negative and you can assume that a BST contains at most 100 elements. A line in which contains the value of -1 indicates the end of input.

# Output

For each test case, print the case number followed by messages indicating the height of the BST and the average of comparisons to search the keys of the BST, in the form:

Case X:

The height of the BST is: Y

The average of the search in the BST is: Z

where X,Y and Z are replaced by the appropriate values. Z could be displayed with 2 decimals.

# Sample Input

10

0 1 2 3 3 5 5 5 5 5

0 0 2 3 3 5 5 5 5 5

0 0 0 3 3 5 5 5 5 5

0 0 0 0 4 5 5 5 5 5

0 0 0 0 0 5 5 5 5 5

0 0 0 0 0 0 6 7 7 9

0 0 0 0 0 0 0 7 7 9

0 0 0 0 0 0 0 0 8 9

0 0 0 0 0 0 0 0 0 9

0 0 0 0 0 0 0 0 0 0

-1

# Sample Output

Case 1:

The height of the BST is: 4

The average of the search in the BST is: 2.89