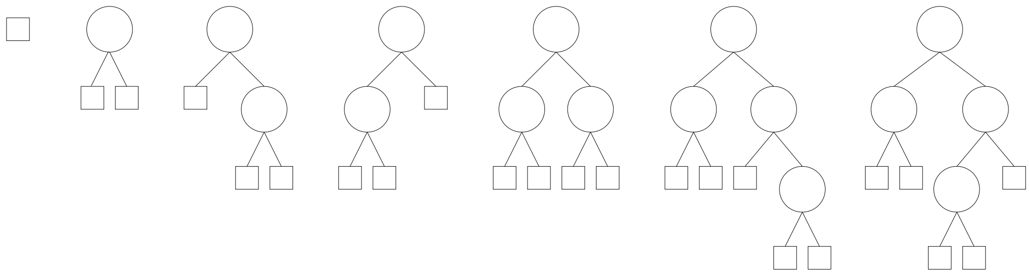
Jutge.org

The Virtual Learning Environment for Computer Programming

Ranking AVL trees **P20941_en**

Tercer Concurs de Programació de la UPC - Semifinal (2005-09-14)

Given a non-empty binary tree T, let T_L and T_R denote respectively the left child of T and the right child of T. A binary tree T is an AVL tree if and only if T is empty, or T_L and T_R are AVL trees such that $|\text{height}(T_L) - \text{height}(T_R)| \leq 1$. These are some examples of AVL trees with respective heights 0, 1, 2, 2, 2, 3 and 3 (a box denotes an empty tree):



We can inductively define a total order over AVL trees as follows: The empty tree is the smallest AVL tree. For every two non-empty AVL trees A and B, $A < B$ if and only if

- $\text{height}(A) < \text{height}(B)$, or
- $\text{height}(A) = \text{height}(B)$ and $A_L < B_L$, or
- $\text{height}(A) = \text{height}(B)$ and $A_L = B_L$ and $A_R < B_R$.

The trees in the picture above are the first, second, ... , seventh AVL trees using this order.

Write a program such that, for every given AVL tree, computes and prints its rank (that is, its position in the infinite sorted list of AVL trees, starting at 1).

Input

Input begins with the number of cases n, followed by n strings, each one with the preorder of an AVL tree, with '`1`' denoting a node and '`0`' denoting a leaf. No given tree has height larger than 6.

Output

For every given AVL tree, print its rank.

Sample input

```
3
100
110010100
11111100010011000111000100111100010011000
```

Sample output

```
2
6
6736354888
```

Problem information

Author : Salvador Roura
Generation : 2013-09-02 15:37:30