ED – exercise 11 – 2024.04.23

Square nodes

Goal

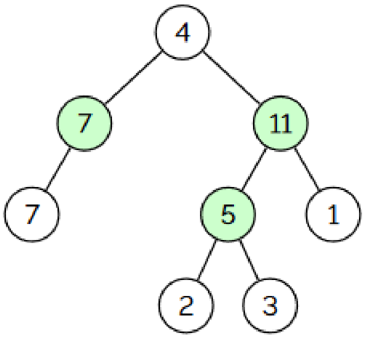
Tree-wrangling. Statement adapted from one of the 2021 final exams.

Instructions

Log into http://ed.fdi.ucm.es/domjudge/team with your credentials. Make sure to **submit at least 1 answer before the period ends**, even if it does not work. Before leaving the classroom, even if your program seems to work, **go by the teacher's desk** to see if it can be improved.

Problem statement

In a binary tree of integers, we will call nodes that have the same value as the sum of their descendants "square nodes". For example, in the given tree, nodes in green are "square" (and all others are not). Write a function that can efficiently locate the deepest square node of such a tree; in case of ties, the leftmost node wins.



Input

Input starts with an integer N: the number of trees to process. Each of the N following lines will contain a tree in the customary format.

Output

For each input tree, output the value of the deepest "square" node in that tree (or -1 if no such node exists).

Example input & output

```
3                                       5
(([7] 7 #) 4 (([2] 5 [3]) 11 [1]))     0
([0] 4 ([2] 3 [1]))                     -1
([1] 3 #)
```

Additional guidance

You ***must*** use the provided template. Also, your answer must be *external* to the BinTree.h structure: you cannot access Node *s directly, nor can you modify the internal implementation of your BinTree. Your answer must be efficient, and you must include a *comment stating the worst-case asymptotic cost of your answer*.