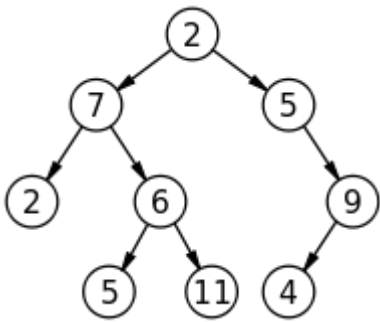


You are given a recursive notation of a binary tree: each node of a tree is represented as a set of three elements:

- value of the node;
- left subtree;
- right subtree.

So, a tree can be written as (value left\_subtree right\_subtree) . It is guaranteed that  $1 \leq \text{value} \leq 10^9$  . If a node doesn't exist then it is represented as an empty set: () . For example, here is a representation of a tree in the given picture:

(2 (7 (2 () ()) (6 (5 () ()) (11 () ()))) (5 () (9 (4 () ()) ())))



Your task is to obtain a list of nodes, that are the most distant from the tree root, in the order from left to right.

In the notation of a node its value and subtrees are separated by exactly one space character.

Example

For

tree = "(2 (7 (2 () ()) (6 (5 () ()) (11 () ()))) (5 () (9 (4 () ()) ())))"

the output should be  
treeBottom(tree) = [5, 11, 4] .

Input/Output

- [execution time limit] 4 seconds (js)
- [input] string tree

Guaranteed constraints:  
 $5 \leq \text{tree.length} \leq 120$  .

- [output] array.integer

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
}
```