B Building Beaver

Time limit: 6s

Beaver Bert is an expert in everything trees: aspen trees, willow trees, poplar trees, birch trees, and *binary search trees* (BSTs). Most of these trees are useful for building dams, but to keep track of the inventory of all felled trees for his dam, Bert uses a BST. A BST is a binary tree with a value in each node. These values satisfy the BST property: the value at each node is strictly larger than all values in its left subtree and strictly smaller than all values in its right subtree.

A simple BST insertion, given a value x that does not yet appear in the tree, goes as follows:

- Start at the root.
- If the value of the current node is smaller than x, recurse to the right subtree. Otherwise, recurse to the left subtree.
- If this subtree does not exist, insert a new node at this position with value x.

The algorithm does not perform rotations to rebalance the tree.

As Beaver Bert already felled some trees, he wants to add them to his inventory. For simplicity, he gives a unique identifier between 1 and n to each of his n felled trees. Beaver Bert adds the felled tree identifiers to the BST one by one by using simple BST insertion by choosing an order $p_1, \ldots, p_n$ of the numbers 1 to n. Specifically, he makes p_1 the root, and then inserts $p_2, p_3, \ldots, p_n$ into the simple BST one by one.

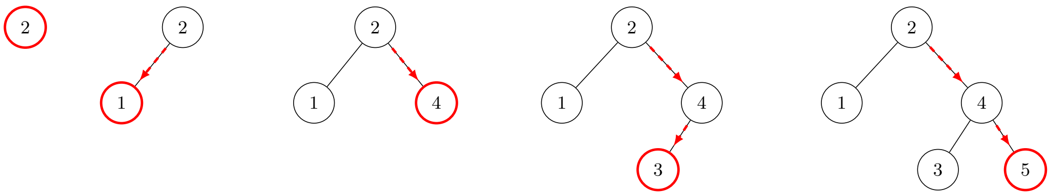

Figure B.1: Example for inserting the values in order $(2, 1, 4, 3, 5)$ in a BST, corresponding to the second sample output.

It would be really nice if, after this process, the BST satisfies the AVL balance property,[1] so Busy Beaver Bert has no problems with looking up specific felled trees in his inventory fast. Help Bert by finding the lexicographically smallest order[2] of the felled tree identifiers $1, \ldots, n$ that makes the BST satisfy the AVL balance property.

---

[1]The AVL balance property of a binary tree states that at each vertex, the difference between the two heights of its children's (possibly absent) subtrees is at most 1. Note that the height of an absent subtree is equal to 0.

[2]A list of numbers $p_1, p_2, \ldots, p_n$ is lexicographically smaller than $q_1, q_2, \ldots, q_n$, if on the first index f where p and q differ, $p_f < q_f$.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 2 \cdot 10^5$), the desired size of the BST.

Output

Output an ordering of the numbers 1 to n, $p_1, \ldots, p_n$, the lexicographically smallest order of the felled trees such that, when inserted into a simple BST in that order, this BST satisfies the AVL property.

It can be proven that some order exists such that the AVL property holds.

Sample Input 1	Sample Output 1
`1`	`1`

Sample Input 2	Sample Output 2
`5`	`2 1 4 3 5`