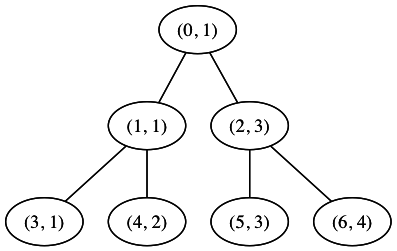
Let’s consider an array of **any** **distinct** **integers** with size . We are building backward the following binary tree: the leaves contain the value of the array with the same order as the original array. A given node on a non-leaf is obtained by taking the minimum value of the subtree.

For instance, let’s consider the array A = [1, 2, 3, 4]. Our tree would be the following, where the first number is the node number and the second number is the node value:



When we look at the node values in order, we obtain the following new array: [1, 1, 3, 1, 2, 3, 4]

The question is the following:

Given an array of size 2n-1, is it possible to shuffle the array and make it correspond to an array that would be obtained using the node values of a tree built like above? If, yes, can you output the corresponding array?

Additionally, if there are multiple solutions, the output would be the smallest array, ordered lexicographically.

Example 1:

Input = [3, 1, 3, 1, 2, 4, 1]

Output = [1, 1, 3, 1, 2, 3, 4] (corresponding to the above tree, original array = [1,2,3,4])

Example 2:

Input = [1, 1, 1]

Output = Impossible. The leaves of the tree would be [1, 1], meaning that the original array is not an array of distinct integers.

The solution can be solved in the language of your choice (whatever you feel is the best language for this particular solution). We will look especially at the insights related to speed and complexity.