Boxes Packing

You are given n rectangular boxes, the i^{th} box has the length $length_i$, the width $width_i$ and the height $height_i$. Your task is to check if it is possible to pack all boxes into one so that inside each box there is no more than one another box (which, in turn, can contain at most one another box, and so on). More formally, your task is to check whether there is such sequence of n different numbers p_i ($1 \le p_i \le n$) that for each $1 \le i < n$ the box number p_i can be put into the box number p_{i+1}.

A box can be put into another box if all sides of the first one are less than the respective ones of the second one. You can rotate each box as you wish, i.e. you can swap its length, width and height if necessary.

Example

- For `length = [1, 3, 2]`, `width = [1, 3, 2]` and `height = [1, 3, 2]`, the output should be `boxesPacking(length, width, height) = true`;
- For `length = [1, 1]`, `width = [1, 1]` and `height = [1, 1]`, the output should be `boxesPacking(length, width, height) = false`;
- For `length = [3, 1, 2]`, `width = [3, 1, 2]` and `height = [3, 2, 1]`, the output should be `boxesPacking(length, width, height) = false`.

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] array.integer length**

  Array of positive integers.

  Guaranteed constraints:

  $1 \le length.length \le 10^4$,

  $1 \le length[i] \le 2 \cdot 10^4$.

- **[input] array.integer width**

  Array of positive integers.

  Guaranteed constraints:

  $width.length = length.length$,

  $1 \le width[i] \le 2 \cdot 10^4$.

- **[input] array.integer height**

  Array of positive integers.

  Guaranteed constraints:

  $height.length = length.length$,

  $1 \le height[i] \le 2 \cdot 10^4$.

- **[output] boolean**

  `true` if it is possible to put all boxes into one, `false` otherwise.

[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;
}
```