2097. Valid Arrangement of Pairs

Description

You are given a **0-indexed** 2D integer array `pairs` where `pairs[i] = [start_i, end_i]`. An arrangement of `pairs` is **valid** if for every index `i` where `1 <= i < pairs.length`, we have `end_{i-1} == start_i`.

Return ***any*** *valid arrangement of* `pairs`.

Note: The inputs will be generated such that there exists a valid arrangement of `pairs`.

Example 1:

```
Input: pairs = [[5,1],[4,5],[11,9],[9,4]]
Output: [[11,9],[9,4],[4,5],[5,1]]
Explanation:
This is a valid arrangement since end_{i-1} always equals start_i.
end_0 = 9 == 9 = start_1
end_1 = 4 == 4 = start_2
end_2 = 5 == 5 = start_3
```

Example 2:

```
Input: pairs = [[1,3],[3,2],[2,1]]
Output: [[1,3],[3,2],[2,1]]
Explanation:
This is a valid arrangement since end_{i-1} always equals start_i.
end_0 = 3 == 3 = start_1
end_1 = 2 == 2 = start_2
The arrangements [[2,1],[1,3],[3,2]] and [[3,2],[2,1],[1,3]] are also valid.
```

Example 3:

```
Input: pairs = [[1,2],[1,3],[2,1]]
Output: [[1,2],[2,1],[1,3]]
Explanation:
This is a valid arrangement since end_{i-1} always equals start_i.
end_0 = 2 == 2 = start_1
end_1 = 1 == 1 = start_2
```

Constraints:

- `1 <= pairs.length <= 10^5`
- `pairs[i].length == 2`
- `0 <= start_i, end_i <= 10^9`
- `start_i != end_i`
- No two pairs are exactly the same.
- There **exists** a valid arrangement of `pairs`.