

5932. Valid Arrangement of Pairs

My Submissions (/contest/weekly-contest-270/problems/valid-arrangement-of-pairs/submissions/)

Back to Contest (/contest/weekly-contest-270/)

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

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 `endi-1` always equals `starti`.
`end0 = 9 == 9 = start1`
`end1 = 4 == 4 = start2`
`end2 = 5 == 5 = start3`

Example 2:

Input: `pairs = [[1,3],[3,2],[2,1]]`
Output: `[[1,3],[3,2],[2,1]]`
Explanation:
This is a valid arrangement since `endi-1` always equals `starti`.
`end0 = 3 == 3 = start1`
`end1 = 2 == 2 = start2`
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 `endi-1` always equals `starti`.
`end0 = 2 == 2 = start1`
`end1 = 1 == 1 = start2`

Constraints:

- `1 <= pairs.length <= 105`
- `pairs[i].length == 2`
- `0 <= starti, endi <= 109`
- `starti != endi`
- No two pairs are exactly the same.
- There **exists** a valid arrangement of `pairs`.

JavaScript

```
1 /**
2  * @param {number[][]} pairs
3  * @return {number[][]}
4  */
5 var validArrangement = function(pairs) {
6
7  };
```