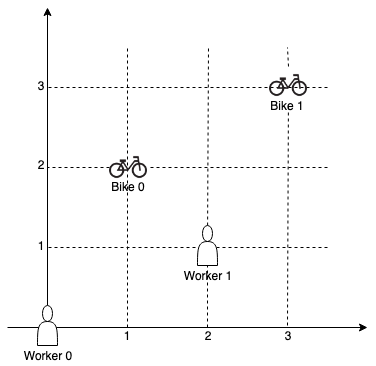
On a campus represented as a 2D grid, there are N workers and M bikes, with N <= M. Each worker and bike is a 2D coordinate on this grid.

Our goal is to assign a bike to each worker. Among the available bikes and workers, we choose the (worker, bike) pair with the shortest Manhattan distance between each other, and assign the bike to that worker. (If there are multiple (worker, bike) pairs with the same shortest Manhattan distance, we choose the pair with the smallest worker index; if there are multiple ways to do that, we choose the pair with the smallest bike index). We repeat this process until there are no available workers.

The Manhattan distance between two points p1 and p2 is Manhattan(p1, p2) = |p1.x - p2.x| + |p1.y - p2.y|.

Return a vector ans of length N, where ans[i] is the index (0-indexed) of the bike that the i-th worker is assigned to.

**Example 1:**



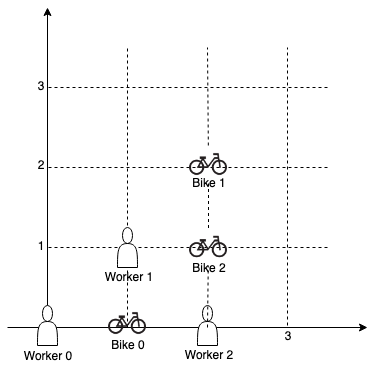
**Input:** workers = [[0,0],[2,1]], bikes = [[1,2],[3,3]]

**Output:** [1,0]

**Explanation:**

Worker 1 grabs Bike 0 as they are closest (without ties), and Worker 0 is assigned Bike 1. So the output is [1, 0].

**Example 2:**



**Input:** workers = [[0,0],[1,1],[2,0]], bikes = [[1,0],[2,2],[2,1]]

**Output:** [0,2,1]

**Explanation:**

Worker 0 grabs Bike 0 at first. Worker 1 and Worker 2 share the same distance to Bike 2, thus Worker 1 is assigned to Bike 2, and Worker 2 will take Bike 1. So the output is [0,2,1].

**Note:**

1. 0 <= workers[i][j], bikes[i][j] < 1000
2. All worker and bike locations are distinct.
3. 1 <= workers.length <= bikes.length <= 1000