**Problem 1: Parcel Sorting**

**Problem Statement:**

A warehouse has nnn parcels that need to be sorted and delivered to the correct locations. The parcels are arranged in a circular conveyor belt, and each parcel has a unique ID. The IDs on the conveyor belt are jumbled and do not match their target positions.

The target arrangement of parcels is given as target[]target[]target[], and the current arrangement of parcels is current[]current[]current[].

You can perform the following operation:

* Swap two parcels on the conveyor belt.

Your task is to determine the **minimum number of swaps** required to rearrange the parcels on the conveyor belt such that the current[] matches the target[].

**Constraints:**

* 1≤n≤1051 \leq n \leq 10^51≤n≤105
* target[]target[]target[] and current[]current[]current[] are permutations of integers from 1 to nnn.

**Input Format:**

* The first line contains an integer nnn, the number of parcels.
* The second line contains nnn integers, representing the target arrangement target[]target[]target[].
* The third line contains nnn integers, representing the current arrangement current[]current[]current[].

**Output Format:**

* A single integer denoting the minimum number of swaps required.

**Example Input:**

5

1 2 3 4 5

4 3 2 1 5

**Example Output:**

2

**Explanation:**

* Swap parcel 4 and parcel 1: [4,3,2,1,5]→[1,3,2,4,5][4, 3, 2, 1, 5] \rightarrow [1, 3, 2, 4, 5][4,3,2,1,5]→[1,3,2,4,5].
* Swap parcel 3 and parcel 2: [1,3,2,4,5]→[1,2,3,4,5][1, 3, 2, 4, 5] \rightarrow [1, 2, 3, 4, 5][1,3,2,4,5]→[1,2,3,4,5].
* Total swaps = 2.

**Problem 2: Drone Delivery Cycles**

**Problem Statement:**

A company uses drones to deliver packages to customers. Each drone has a fixed route, but due to some errors in the route planning system, certain delivery points are swapped between drones.

The company wants to optimize the delivery by ensuring that each delivery point is served by the correct drone. This requires swapping delivery points between drones until all routes are correct.

The routes are represented as a directed graph where:

* Each delivery point is a node.
* A directed edge u→vu \to vu→v means delivery point uuu is currently served by the drone responsible for vvv.

Your task is to:

1. Detect **cycles** in the delivery routes where delivery points are incorrectly assigned.
2. Determine the **minimum number of swaps** needed to resolve the incorrect routes.

**Constraints:**

* 1≤n≤1051 \leq n \leq 10^51≤n≤105, where nnn is the number of delivery points.
* Each delivery point appears exactly once in the route.

**Input Format:**

* The first line contains an integer nnn, the number of delivery points.
* The second line contains nnn integers, where the iii-th integer represents the delivery point p[i]p[i]p[i], such that p[i]=vp[i] = vp[i]=v means delivery point iii is currently assigned to vvv.

**Output Format:**

* A single integer denoting the minimum number of swaps required to resolve all delivery routes.

**Example Input:**

5

2 3 1 5 4

**Example Output:**

2

**Explanation:**

* The delivery routes form a cycle: 1→2→3→11 \to 2 \to 3 \to 11→2→3→1, and 4→5→44 \to 5 \to 44→5→4.
* Two swaps are needed to resolve the routes:
  + Swap 111 and 333: [2,3,1,5,4]→[1,3,2,5,4][2, 3, 1, 5, 4] \to [1, 3, 2, 5, 4][2,3,1,5,4]→[1,3,2,5,4].
  + Swap 444 and 555: [1,3,2,5,4]→[1,2,3,4,5][1, 3, 2, 5, 4] \to [1, 2, 3, 4, 5][1,3,2,5,4]→[1,2,3,4,5].