Problem 765: Couples Holding Hands

Problem Information

Difficulty: Hard
Acceptance Rate: 58.93%
Paid Only: No
Tags: Greedy, Depth-First Search, Breadth-First Search, Union Find, Graph

Problem Description

There are `n` couples sitting in `2n` seats arranged in a row and want to hold hands.

The people and seats are represented by an integer array `row` where `row[i]` is the ID of the person sitting in the `ith` seat. The couples are numbered in order, the first couple being `(0, 1)`, the second couple being `(2, 3)`, and so on with the last couple being `(2n - 2, 2n - 1)`.

Return _the minimum number of swaps so that every couple is sitting side by side_. A swap consists of choosing any two people, then they stand up and switch seats.

Example 1:

Input: row = [0,2,1,3] **Output:** 1 **Explanation:** We only need to swap the second (row[1]) and third (row[2]) person.

Example 2:

Input: row = [3,2,0,1] **Output:** 0 **Explanation:** All couples are already seated side by side.

Constraints:

* `2n == row.length` * `2 <= n <= 30` * `0 <= row[i] < 2n` * All the elements of `row` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
int minSwapsCouples(vector<int>& row) {

}
};
```

Java:

```
class Solution {
public int minSwapsCouples(int[] row) {

}
}
```

Python3:

```
class Solution:
def minSwapsCouples(self, row: List[int]) -> int:
```