Problem 849: Maximize Distance to Closest Person

Problem Information

Difficulty: Medium
Acceptance Rate: 49.42%
Paid Only: No
Tags: Array

Problem Description

You are given an array representing a row of `seats` where `seats[i] = 1` represents a person sitting in the `ith` seat, and `seats[i] = 0` represents that the `ith` seat is empty **(0-indexed)**.

There is at least one empty seat, and at least one person sitting.

Alex wants to sit in the seat such that the distance between him and the closest person to him is maximized.

Return _that maximum distance to the closest person_.

Example 1:

![](https://assets.leetcode.com/uploads/2020/09/10/distance.jpg)

Input: seats = [1,0,0,0,1,0,1] **Output:** 2 **Explanation:** If Alex sits in the second open seat (i.e. seats[2]), then the closest person has distance 2. If Alex sits in any other open seat, the closest person has distance 1. Thus, the maximum distance to the closest person is 2.

Example 2:

Input: seats = [1,0,0,0] **Output:** 3 **Explanation:** If Alex sits in the last seat (i.e. seats[3]), the closest person is 3 seats away. This is the maximum distance possible, so the answer is 3.

Example 3:

Input: seats = [0,1] **Output:** 1

Constraints:

* `2 <= seats.length <= 2 * 104` * `seats[i]` is `0` or `1`. * At least one seat is **empty**. * At least one seat is **occupied**.

Code Snippets

C++:

```
class Solution {
public:
int maxDistToClosest(vector<int>& seats) {

}
};
```

Java:

```
class Solution {
public int maxDistToClosest(int[] seats) {

}
}
```

Python3:

```
class Solution:
def maxDistToClosest(self, seats: List[int]) -> int:
```