Problem 2511: Maximum Enemy Forts That Can Be Captured

Problem Information

Difficulty: Easy
Acceptance Rate: 40.70%
Paid Only: No
Tags: Array, Two Pointers

Problem Description

You are given a **0-indexed** integer array `forts` of length `n` representing the positions of several forts. `forts[i]` can be `-1`, `0`, or `1` where:

* `-1` represents there is **no fort** at the `ith` position. * `0` indicates there is an **enemy** fort at the `ith` position. * `1` indicates the fort at the `ith` the position is under your command.

Now you have decided to move your army from one of your forts at position `i` to an empty position `j` such that:

* `0 <= i, j <= n - 1` * The army travels over enemy forts **only**. Formally, for all `k` where `min(i,j) < k < max(i,j)`, `forts[k] == 0.`

While moving the army, all the enemy forts that come in the way are **captured**.

Return _the**maximum** number of enemy forts that can be captured_. In case it is **impossible** to move your army, or you do not have any fort under your command, return `0` _._

Example 1:

Input: forts = [1,0,0,-1,0,0,0,0,1] **Output:** 4 **Explanation:** - Moving the army from position 0 to position 3 captures 2 enemy forts, at 1 and 2. - Moving the army from position 8 to position 3 captures 4 enemy forts. Since 4 is the maximum number of enemy forts that can be captured, we return 4.

Example 2:

Input: forts = [0,0,1,-1] **Output:** 0 **Explanation:** Since no enemy fort can be captured, 0 is returned.

Constraints:

* `1 <= forts.length <= 1000` * `-1 <= forts[i] <= 1`

Code Snippets

C++:

```
class Solution {
public:
int captureForts(vector<int>& forts) {

}
};
```

Java:

```
class Solution {
public int captureForts(int[] forts) {

}
}
```

Python3:

```
class Solution:
def captureForts(self, forts: List[int]) -> int:
```