2511. Maximum Enemy Forts That Can Be Captured

My Submissions (/contest/biweekly-contest-94/problems/maximum-enemy-forts-that-can-be-captured/submissions/)

Back to Contest (/contest/biweekly-contest-94/)

User Accepted:	6959
User Tried:	8777
Total Accepted:	7168
Total Submissions:	23674
Difficulty:	Easy

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 `i`th position.
- `0` indicates there is an **enemy** fort at the `i`th position.
- `1` indicates the fort at the `i`th 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`

Discuss (https://leetcode.com/problems/maximum-enemy-forts-that-can-be-captured/discuss)

JavaScript

```
const cutMaxConsecutive = (as) => { let d = [], l = 0, n = as.length; for (let i = 0; i + 1 < n; i++) { if (as[i + 1] != as[i]) { d.push(as.slice(l, i + 1)); l = i + 1; } } d.push(as.slice(l)); return d; };

const captureForts = function(a) {
    let d = cutMaxConsecutive(a), res = 0;
    for (let i = 1; i < d.length - 1; i++) {
        if (d[i][0] == 0 && d[i-1][0] * d[i+1][0] == -1) res = Math.max(res, d[i].length);
    }
    return res;
};
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/865209977/) More Details (/submissions/detail/865209977/)

Share your acceptance!

11

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)