Problem 1989: Maximum Number of People That Can Be Caught in Tag

Problem Information

Difficulty: Medium
Acceptance Rate: 49.40%
Paid Only: Yes
Tags: Array, Greedy

Problem Description

You are playing a game of tag with your friends. In tag, people are divided into two teams: people who are "it", and people who are not "it". The people who are "it" want to catch as many people as possible who are not "it".

You are given a **0-indexed** integer array `team` containing only zeros (denoting people who are **not** "it") and ones (denoting people who are "it"), and an integer `dist`. A person who is "it" at index `i` can catch any **one** person whose index is in the range `[i - dist, i + dist]` (**inclusive**) and is **not** "it".

Return _the**maximum** number of people that the people who are "it" can catch_.

Example 1:

Input: team = [0,1,0,1,0], dist = 3 **Output:** 2 **Explanation:** The person who is "it" at index 1 can catch people in the range [i-dist, i+dist] = [1-3, 1+3] = [-2, 4]. They can catch the person who is not "it" at index 2. The person who is "it" at index 3 can catch people in the range [i-dist, i+dist] = [3-3, 3+3] = [0, 6]. They can catch the person who is not "it" at index 0. The person who is not "it" at index 4 will not be caught because the people at indices 1 and 3 are already catching one person.

Example 2:

Input: team = [1], dist = 1 **Output:** 0 **Explanation:** There are no people who are not "it" to catch.

Example 3:

Input: team = [0], dist = 1 **Output:** 0 **Explanation:** There are no people who are "it" to catch people.

Constraints:

* `1 <= team.length <= 105` * `0 <= team[i] <= 1` * `1 <= dist <= team.length`

Code Snippets

C++:

```cpp
class Solution {
public:
    int catchMaximumAmountofPeople(vector<int>& team, int dist) {


    }
};
```

Java:

```java
class Solution {
    public int catchMaximumAmountofPeople(int[] team, int dist) {


    }
}
```

Python3:

```python
class Solution:
    def catchMaximumAmountofPeople(self, team: List[int], dist: int) -> int:
```