Problem 957: Prison Cells After N Days

Problem Information

Difficulty: Medium
Acceptance Rate: 39.04%
Paid Only: No
Tags: Array, Hash Table, Math, Bit Manipulation

Problem Description

There are `8` prison cells in a row and each cell is either occupied or vacant.

Each day, whether the cell is occupied or vacant changes according to the following rules:

* If a cell has two adjacent neighbors that are both occupied or both vacant, then the cell becomes occupied. * Otherwise, it becomes vacant.

Note that because the prison is a row, the first and the last cells in the row can't have two adjacent neighbors.

You are given an integer array `cells` where `cells[i] == 1` if the `ith` cell is occupied and `cells[i] == 0` if the `ith` cell is vacant, and you are given an integer `n`.

Return the state of the prison after `n` days (i.e., `n` such changes described above).

Example 1:

Input: cells = [0,1,0,1,1,0,0,1], n = 7 **Output:** [0,0,1,1,0,0,0,0] **Explanation:** The following table summarizes the state of the prison on each day: Day 0: [0, 1, 0, 1, 1, 0, 0, 1] Day 1: [0, 1, 1, 0, 0, 0, 0, 0] Day 2: [0, 0, 0, 0, 1, 1, 1, 0] Day 3: [0, 1, 1, 0, 0, 1, 0, 0] Day 4: [0, 0, 0, 0, 0, 1, 0, 0] Day 5: [0, 1, 1, 1, 0, 1, 0, 0] Day 6: [0, 0, 1, 0, 1, 1, 0, 0] Day 7: [0, 0, 1, 1, 0, 0, 0, 0]

Example 2:

Input: cells = [1,0,0,1,0,0,1,0], n = 1000000000 **Output:** [0,0,1,1,1,1,1,0]

Constraints:

* `cells.length == 8` * `cells[i]` is either `0` or `1`. * `1 <= n <= 109`

Code Snippets

C++:

```
class Solution {
public:
vector<int> prisonAfterNDays(vector<int>& cells, int n) {

}
};
```

Java:

```
class Solution {
public int[] prisonAfterNDays(int[] cells, int n) {

}
}
```

Python3:

```
class Solution:
def prisonAfterNDays(self, cells: List[int], n: int) -> List[int]:
```