Prison Cells After N Days

*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, the changes are 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.)***

We describe the current state of the prison in the following way: cells[ i ] = 1 and the ith cell is occupied, else cells[ i ] = 0.

Given the initial state of the prison, return the state of the prison after N days (and N such changes described above.)

*Example 1:*

Input: cells = [0, 1, 0, 1, 1, 0, 0, 1], N = 7

Output: cells = [0, 0, 1, 1, 0, 0, 0, 0]

Explanation:

The following summarizes the state of the prison on each day:

**Day 0: cells = [0, 1, 0, 1, 1, 0, 0, 1] – Original Cells**

**Day 1: cells = [0, 1, 1, 0, 0, 0, 0, 0]**

Day 2: cells = [0, 0, 0, 0, 1, 1, 1, 0]

Day 3: cells = [0, 1, 1, 0, 0, 1, 0, 0]

Day 4: cells = [0, 0, 0, 0, 0, 1, 0, 0]

Day 5: cells = [0, 1, 1, 1, 0, 1, 0, 0]

Day 6: cells = [0, 0, 1, 0, 1, 1, 0, 0]

Day 7: cells = [0, 0, 1, 1, 0, 0, 0, 0]

Day 8: cells = [0, 0, 0, 0, 0, 1, 1, 0]

Day 9: cells = [0, 1, 1, 1, 0, 0, 0, 0]

Day 10: cells = [0, 0, 1, 0, 0, 1, 1, 0]

Day 11: cells = [0, 0, 1, 0, 0, 0, 0, 0]

Day 12: cells = [0, 0, 1, 0, 1, 1, 1, 0]

Day 13: cells = [0, 0, 1, 1, 0, 1, 0, 0]

Day 14: cells = [0, 0, 0, 0, 1, 1, 0, 0]

**Day 15: cells = [0, 1, 1, 0, 0, 0, 0, 0]**

Day 16: cells = [0, 0, 0, 0, 1, 1, 1, 0]

*Conclusion:*

Deduce from the calculation, we can conclude that for each 14 times/round, we would get one circulation. Therefore, we can use N to mode 14 and get the specific day with the same value cells.

*Code:*

*class Solution {*

*public:*

*vector<int> prisonAfterNDays(vector<int>& cells, int N) {*

*if (N == 0) return cells;*

*N = N % 14;*

*if (N == 0) N = 14;*

*for (int i = 1; i <= N; i++) {*

*vector<int> intercells(cells);*

*for (int j = 0; j < 8; j++) {*

*if (j < 1 || j > 6) {*

*cells[j] = 0;*

*} else if (j >= 1 && j <= 6) {*

*cells[j] = intercells[j - 1] ^ !intercells[j + 1];*

*}*

*}*

*}*

*return cells;*

*}*

*};*

*Reference:*

1. Leetcode user [wpn-zju](https://leetcode-cn.com/u/wpn-zju/).