Problem 3495: Minimum Operations to Make Array Elements Zero

Problem Information

Difficulty: Hard
Acceptance Rate: 60.38%
Paid Only: No
Tags: Array, Math, Bit Manipulation

Problem Description

You are given a 2D array `queries`, where `queries[i]` is of the form `[l, r]`. Each `queries[i]` defines an array of integers `nums` consisting of elements ranging from `l` to `r`, both **inclusive**.

In one operation, you can:

* Select two integers `a` and `b` from the array. * Replace them with `floor(a / 4)` and `floor(b / 4)`.

Your task is to determine the **minimum** number of operations required to reduce all elements of the array to zero for each query. Return the sum of the results for all queries.

Example 1:

Input: queries = [[1,2],[2,4]]

Output: 3

Explanation:

For `queries[0]`:

* The initial array is `nums = [1, 2]`. * In the first operation, select `nums[0]` and `nums[1]`. The array becomes `[0, 0]`. * The minimum number of operations required is 1.

For `queries[1]`:

* The initial array is `nums = [2, 3, 4]`. * In the first operation, select `nums[0]` and `nums[2]`. The array becomes `[0, 3, 1]`. * In the second operation, select `nums[1]` and `nums[2]`. The array becomes `[0, 0, 0]`. * The minimum number of operations required is 2.

The output is `1 + 2 = 3`.

Example 2:

Input: queries = [[2,6]]

Output: 4

Explanation:

For `queries[0]`:

* The initial array is `nums = [2, 3, 4, 5, 6]`. * In the first operation, select `nums[0]` and `nums[3]`. The array becomes `[0, 3, 4, 1, 6]`. * In the second operation, select `nums[2]` and `nums[4]`. The array becomes `[0, 3, 1, 1, 1]`. * In the third operation, select `nums[1]` and `nums[2]`. The array becomes `[0, 0, 0, 1, 1]`. * In the fourth operation, select `nums[3]` and `nums[4]`. The array becomes `[0, 0, 0, 0, 0]`. * The minimum number of operations required is 4.

The output is 4.

Constraints:

* `1 <= queries.length <= 105` * `queries[i].length == 2` * `queries[i] == [l, r]` * `1 <= l < r <= 109`

Code Snippets

C++:

```
class Solution {
public:
```

```
long long minOperations(vector<vector<int>>& queries) {

}
};
```

Java:

```
class Solution {
public long minOperations(int[][] queries) {

}
}
```

Python3:

```
class Solution:
def minOperations(self, queries: List[List[int]]) -> int:
```