Problem 2438: Range Product Queries of Powers

Problem Information

Difficulty: Medium
Acceptance Rate: 61.37%
Paid Only: No
Tags: Array, Bit Manipulation, Prefix Sum

Problem Description

Given a positive integer `n`, there exists a **0-indexed** array called `powers`, composed of the **minimum** number of powers of `2` that sum to `n`. The array is sorted in **non-decreasing** order, and there is **only one** way to form the array.

You are also given a **0-indexed** 2D integer array `queries`, where `queries[i] = [lefti, righti]`. Each `queries[i]` represents a query where you have to find the product of all `powers[j]` with `lefti <= j <= righti`.

Return _an array_`answers` _, equal in length to_`queries` _, where_`answers[i]`_is the answer to the_`ith` _query_. Since the answer to the `ith` query may be too large, each `answers[i]` should be returned **modulo** `109 + 7`.

Example 1:

Input: n = 15, queries = [[0,1],[2,2],[0,3]] **Output:** [2,4,64] **Explanation:** For n = 15, powers = [1,2,4,8]. It can be shown that powers cannot be a smaller size. Answer to 1st query: powers[0] * powers[1] = 1 * 2 = 2. Answer to 2nd query: powers[2] = 4. Answer to 3rd query: powers[0] * powers[1] * powers[2] * powers[3] = 1 * 2 * 4 * 8 = 64. Each answer modulo 109 + 7 yields the same answer, so [2,4,64] is returned.

Example 2:

Input: n = 2, queries = [[0,0]] **Output:** [2] **Explanation:** For n = 2, powers = [2]. The answer to the only query is powers[0] = 2. The answer modulo 109 + 7 is the same, so [2] is returned.

Constraints:

* `1 <= n <= 109` * `1 <= queries.length <= 105` * `0 <= starti <= endi < powers.length`

Code Snippets

C++:

```
class Solution {
public:
vector<int> productQueries(int n, vector<vector<int>>& queries) {

}
};
```

Java:

```
class Solution {
public int[] productQueries(int n, int[][] queries) {

}
}
```

Python3:

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