Problem 3344: Maximum Sized Array

Problem Information

Difficulty: Medium
Acceptance Rate: 51.32%
Paid Only: Yes
Tags: Binary Search, Bit Manipulation

Problem Description

Given a positive integer `s`, let `A` be a 3D array of dimensions `n × n × n`, where each element `A[i][j][k]` is defined as:

* `A[i][j][k] = i * (j OR k)`, where `0 <= i, j, k < n`.

Return the **maximum** possible value of `n` such that the **sum** of all elements in array `A` does not exceed `s`.

Example 1:

Input: s = 10

Output: 2

Explanation:

* Elements of the array `A` for `n = 2`**:** * `A[0][0][0] = 0 * (0 OR 0) = 0` * `A[0][0][1] = 0 * (0 OR 1) = 0` * `A[0][1][0] = 0 * (1 OR 0) = 0` * `A[0][1][1] = 0 * (1 OR 1) = 0` * `A[1][0][0] = 1 * (0 OR 0) = 0` * `A[1][0][1] = 1 * (0 OR 1) = 1` * `A[1][1][0] = 1 * (1 OR 0) = 1` * `A[1][1][1] = 1 * (1 OR 1) = 1` * The total sum of the elements in array `A` is 3, which does not exceed 10, so the maximum possible value of `n` is 2.

Example 2:

Input: s = 0

Output: 1

Explanation:

* Elements of the array `A` for `n = 1`: * `A[0][0][0] = 0 * (0 OR 0) = 0` * The total sum of the elements in array `A` is 0, which does not exceed 0, so the maximum possible value of `n` is 1.

Constraints:

* `0 <= s <= 1015`

Code Snippets

C++:

```
class Solution {
public:
int maxSizedArray(long long s) {

}
};
```

Java:

```
class Solution {
public int maxSizedArray(long s) {

}
}
```

Python3:

```
class Solution:
def maxSizedArray(self, s: int) -> int:
```