Problem 2557: Maximum Number of Integers to Choose From a Range II

Problem Information

Difficulty: Medium
Acceptance Rate: 34.98%
Paid Only: Yes
Tags: Array, Binary Search, Greedy, Sorting

Problem Description

You are given an integer array `banned` and two integers `n` and `maxSum`. You are choosing some number of integers following the below rules:

* The chosen integers have to be in the range `[1, n]`. * Each integer can be chosen **at most once**. * The chosen integers should not be in the array `banned`. * The sum of the chosen integers should not exceed `maxSum`.

Return _the**maximum** number of integers you can choose following the mentioned rules_.

Example 1:

Input: banned = [1,4,6], n = 6, maxSum = 4 **Output:** 1 **Explanation:** You can choose the integer 3. 3 is in the range [1, 6], and do not appear in banned. The sum of the chosen integers is 3, which does not exceed maxSum.

Example 2:

Input: banned = [4,3,5,6], n = 7, maxSum = 18 **Output:** 3 **Explanation:** You can choose the integers 1, 2, and 7. All these integers are in the range [1, 7], all do not appear in banned, and their sum is 10, which does not exceed maxSum.

Constraints:

* `1 <= banned.length <= 105` * `1 <= banned[i] <= n <= 109` * `1 <= maxSum <= 1015`

Code Snippets

C++:

```
class Solution {
public:
int maxCount(vector<int>& banned, int n, long long maxSum) {

}
};
```

Java:

```
class Solution {
public int maxCount(int[] banned, int n, long maxSum) {

}
}
```

Python3:

```
class Solution:
def maxCount(self, banned: List[int], n: int, maxSum: int) -> int:
```