Problem 3155: Maximum Number of Upgradable Servers

Problem Information

Difficulty: Medium
Acceptance Rate: 42.34%
Paid Only: Yes
Tags: Array, Math, Binary Search

Problem Description

You have `n` data centers and need to upgrade their servers.

You are given four arrays `count`, `upgrade`, `sell`, and `money` of length `n`, which show:

* The number of servers * The cost of upgrading a single server * The money you get by selling a server * The money you initially have

for each data center respectively.

Return an array `answer`, where for each data center, the corresponding element in `answer` represents the **maximum** number of servers that can be upgraded.

Note that the money from one data center **cannot** be used for another data center.

Example 1:

Input: count = [4,3], upgrade = [3,5], sell = [4,2], money = [8,9]

Output: [3,2]

Explanation:

For the first data center, if we sell one server, we'll have `8 + 4 = 12` units of money and we can upgrade the remaining 3 servers.

For the second data center, if we sell one server, we'll have `9 + 2 = 11` units of money and we can upgrade the remaining 2 servers.

Example 2:

Input: count = [1], upgrade = [2], sell = [1], money = [1]

Output: [0]

Constraints:

* `1 <= count.length == upgrade.length == sell.length == money.length <= 105` * `1 <= count[i], upgrade[i], sell[i], money[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
vector<int> maxUpgrades(vector<int>& count, vector<int>& upgrade,
vector<int>& sell, vector<int>& money) {

}
};
```

Java:

```
class Solution {
public int[] maxUpgrades(int[] count, int[] upgrade, int[] sell, int[] money)
{

}
}
```

Python3:

```
class Solution:
def maxUpgrades(self, count: List[int], upgrade: List[int], sell: List[int],
money: List[int]) -> List[int]:
```