Problem 1981: Minimize the Difference Between Target and Chosen Elements

Problem Information

Difficulty: Medium
Acceptance Rate: 36.37%
Paid Only: No
Tags: Array, Dynamic Programming, Matrix

Problem Description

You are given an `m x n` integer matrix `mat` and an integer `target`.

Choose one integer from **each row** in the matrix such that the **absolute difference** between `target` and the **sum** of the chosen elements is **minimized**.

Return _the**minimum absolute difference**_.

The **absolute difference** between two numbers `a` and `b` is the absolute value of `a - b`.

Example 1:

![](https://assets.leetcode.com/uploads/2021/08/03/matrix1.png)

Input: mat = [[1,2,3],[4,5,6],[7,8,9]], target = 13 **Output:** 0 **Explanation:** One possible choice is to: - Choose 1 from the first row. - Choose 5 from the second row. - Choose 7 from the third row. The sum of the chosen elements is 13, which equals the target, so the absolute difference is 0.

Example 2:

![](https://assets.leetcode.com/uploads/2021/08/03/matrix1-1.png)

Input: mat = [[1],[2],[3]], target = 100 **Output:** 94 **Explanation:** The best possible choice is to: - Choose 1 from the first row. - Choose 2 from the second row. - Choose 3 from the third row. The sum of the chosen elements is 6, and the absolute difference is 94.

Example 3:

![](https://assets.leetcode.com/uploads/2021/08/03/matrix1-3.png)

Input: mat = [[1,2,9,8,7]], target = 6 **Output:** 1 **Explanation:** The best choice is to choose 7 from the first row. The absolute difference is 1.

Constraints:

* `m == mat.length` * `n == mat[i].length` * `1 <= m, n <= 70` * `1 <= mat[i][j] <= 70` * `1 <= target <= 800`

Code Snippets

C++:

```
class Solution {
public:
int minimizeTheDifference(vector<vector<int>>& mat, int target) {

}
};
```

Java:

```
class Solution {
public int minimizeTheDifference(int[][] mat, int target) {

}
}
```

Python3:

```
class Solution:
def minimizeTheDifference(self, mat: List[List[int]], target: int) -> int:
```