Problem 1058: Minimize Rounding Error to Meet Target

Problem Information

Difficulty: Medium
Acceptance Rate: 45.71%
Paid Only: Yes
Tags: Array, Math, String, Greedy, Sorting

Problem Description

Given an array of `prices` `[p1,p2...,pn]` and a `target`, round each price `pi` to `Roundi(pi)` so that the rounded array `[Round1(p1),Round2(p2)...,Roundn(pn)]` sums to the given `target`. Each operation `Roundi(pi)` could be either `Floor(pi)` or `Ceil(pi)`.

Return the string `"-1"` if the rounded array is impossible to sum to `target`. Otherwise, return the smallest rounding error, which is defined as `Σ |Roundi(pi) - (pi)|` for `i` from `1` to `n`, as a string with three places after the decimal.

Example 1:

Input: prices = ["0.700","2.800","4.900"], target = 8 **Output:** "1.000" **Explanation:** Use Floor, Ceil and Ceil operations to get (0.7 - 0) + (3 - 2.8) + (5 - 4.9) = 0.7 + 0.2 + 0.1 = 1.0 .

Example 2:

Input: prices = ["1.500","2.500","3.500"], target = 10 **Output:** "-1" **Explanation:** It is impossible to meet the target.

Example 3:

Input: prices = ["1.500","2.500","3.500"], target = 9 **Output:** "1.500"

Constraints:

* `1 <= prices.length <= 500` * Each string `prices[i]` represents a real number in the range `[0.0, 1000.0]` and has exactly 3 decimal places. * `0 <= target <= 106`

Code Snippets

C++:

```
class Solution {
public:
string minimizeError(vector<string>& prices, int target) {

}
};
```

Java:

```
class Solution {
public String minimizeError(String[] prices, int target) {

}
}
```

Python3:

```
class Solution:
def minimizeError(self, prices: List[str], target: int) -> str:
```