Problem 3568: Minimum Moves to Clean the Classroom

Problem Information

Difficulty: Medium
Acceptance Rate: 26.07%
Paid Only: No
Tags: Array, Hash Table, Bit Manipulation, Breadth-First Search, Matrix

Problem Description

You are given an `m x n` grid `classroom` where a student volunteer is tasked with cleaning up litter scattered around the room. Each cell in the grid is one of the following:

* `'S'`: Starting position of the student * `'L'`: Litter that must be collected (once collected, the cell becomes empty) * `'R'`: Reset area that restores the student's energy to full capacity, regardless of their current energy level (can be used multiple times) * `'X'`: Obstacle the student cannot pass through * `'.'`: Empty space

You are also given an integer `energy`, representing the student's maximum energy capacity. The student starts with this energy from the starting position `'S'`.

Each move to an adjacent cell (up, down, left, or right) costs 1 unit of energy. If the energy reaches 0, the student can only continue if they are on a reset area `'R'`, which resets the energy to its **maximum** capacity `energy`.

Return the **minimum** number of moves required to collect all litter items, or `-1` if it's impossible.

Example 1:

Input: classroom = ["S.", "XL"], energy = 2

Output: 2

Explanation:

* The student starts at cell `(0, 0)` with 2 units of energy. * Since cell `(1, 0)` contains an obstacle 'X', the student cannot move directly downward. * A valid sequence of moves to collect all litter is as follows: * Move 1: From `(0, 0)` -> `(0, 1)` with 1 unit of energy and 1 unit remaining. * Move 2: From `(0, 1)` -> `(1, 1)` to collect the litter `'L'`. * The student collects all the litter using 2 moves. Thus, the output is 2.

Example 2:

Input: classroom = ["LS", "RL"], energy = 4

Output: 3

Explanation:

* The student starts at cell `(0, 1)` with 4 units of energy. * A valid sequence of moves to collect all litter is as follows: * Move 1: From `(0, 1)` -> `(0, 0)` to collect the first litter `'L'` with 1 unit of energy used and 3 units remaining. * Move 2: From `(0, 0)` -> `(1, 0)` to `'R'` to reset and restore energy back to 4. * Move 3: From `(1, 0)` -> `(1, 1)` to collect the second litter `'L'`. * The student collects all the litter using 3 moves. Thus, the output is 3.

Example 3:

Input: classroom = ["L.S", "RXL"], energy = 3

Output: -1

Explanation:

No valid path collects all `'L'`.

Constraints:

* `1 <= m == classroom.length <= 20` * `1 <= n == classroom[i].length <= 20` * `classroom[i][j]` is one of `'S'`, `'L'`, `'R'`, `'X'`, or `'.'` * `1 <= energy <= 50` * There is exactly **one** `'S'` in the grid. * There are **at most** 10 `'L'` cells in the grid.

Code Snippets

C++:

```
class Solution {
public:
int minMoves(vector<string>& classroom, int energy) {

}
};
```

Java:

```
class Solution {
public int minMoves(String[] classroom, int energy) {

}
}
```

Python3:

```
class Solution:
def minMoves(self, classroom: List[str], energy: int) -> int:
```