Problem 1730: Shortest Path to Get Food

Problem Information

Difficulty: Medium
Acceptance Rate: 56.97%
Paid Only: Yes
Tags: Array, Breadth-First Search, Matrix

Problem Description

You are starving and you want to eat food as quickly as possible. You want to find the shortest path to arrive at any food cell.

You are given an `m x n` character matrix, `grid`, of these different types of cells:

* `'*'` is your location. There is **exactly one**`'*'` cell. * `'#'` is a food cell. There may be **multiple** food cells. * `'O'` is free space, and you can travel through these cells. * `'X'` is an obstacle, and you cannot travel through these cells.

You can travel to any adjacent cell north, east, south, or west of your current location if there is not an obstacle.

Return _the**length** of the shortest path for you to reach **any** food cell_. If there is no path for you to reach food, return `-1`.

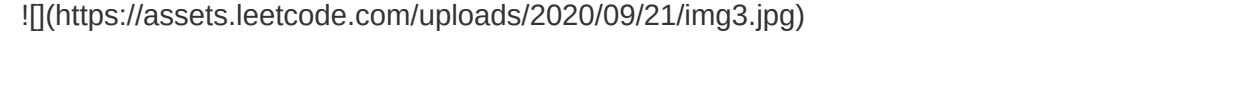
Example 1:

Input: grid = [["X","X","X","X","X","X"],["X","*","O","O","O","X"],["X","O","O","#","O","X"],["X","X","X","X","X","X"]] **Output:** 3 **Explanation:** It takes 3 steps to reach the food.

Example 2:

Input: grid =
[["X","X","X","X","X"],["X","*","X","O","X"],["X","O","X","#","X"],["X","X","X","X","X"]] **Output:**
-1 **Explanation:** It is not possible to reach the food.

Example 3:



Input: grid = [["X","X","X","X","X","X","X","X"],["X","*","O","X","O","#","O","X"],["X","O","O","X","O","O","X","X"],["X","O","O","O","O","#","O","X"],["X","X","X","X","X","X","X","X"]] **Output:**
6 **Explanation:** There can be multiple food cells. It only takes 6 steps to reach the bottom food.

Example 4:

Input: grid = [["X","X","X","X","X","X","X","X"],["X","*","O","X","O","#","O","X"],["X","O","O","X","O","O","X","X"],["X","O","O","O","O","#","O","X"],["O","O","O","O","O","O","O","O"]]
Output: 5

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 200` * `grid[row][col]` is `'*'`, `'X'`, `'O'`, or `'#'`. * The `grid` contains **exactly one** `'*'`.

Code Snippets

C++:

```
class Solution {
public:
int getFood(vector<vector<char>>& grid) {

}
};
```

Java:

```
class Solution {
public int getFood(char[][] grid) {
```

```
}
}
```

Python3:

```
class Solution:
def getFood(self, grid: List[List[str]]) -> int:
```