Problem 864: Shortest Path to Get All Keys

Problem Information

Difficulty: Hard
Acceptance Rate: 54.20%
Paid Only: No
Tags: Array, Bit Manipulation, Breadth-First Search, Matrix

Problem Description

You are given an `m x n` grid `grid` where:

* `'.'` is an empty cell. * `'#'` is a wall. * `'@'` is the starting point. * Lowercase letters represent keys. * Uppercase letters represent locks.

You start at the starting point and one move consists of walking one space in one of the four cardinal directions. You cannot walk outside the grid, or walk into a wall.

If you walk over a key, you can pick it up and you cannot walk over a lock unless you have its corresponding key.

For some `1 <= k <= 6`, there is exactly one lowercase and one uppercase letter of the first `k` letters of the English alphabet in the grid. This means that there is exactly one key for each lock, and one lock for each key; and also that the letters used to represent the keys and locks were chosen in the same order as the English alphabet.

Return _the lowest number of moves to acquire all keys_. If it is impossible, return `-1`.

Example 1:

Input: grid = ["@.a..","###.#","b.A.B"] **Output:** 8 **Explanation:** Note that the goal is to obtain all the keys not to open all the locks.

Example 2:

Input: grid = ["@..aA","..B#.","....b"] **Output:** 6

Example 3:

Input: grid = ["@Aa"] **Output:** -1

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 30` * `grid[i][j]` is either an English letter, `'.'`, `'#'`, or `'@'`. * There is exactly one `'@'` in the grid. * The number of keys in the grid is in the range `[1, 6]`. * Each key in the grid is **unique**. * Each key in the grid has a matching lock.

Code Snippets

C++:

```
class Solution {
public:
int shortestPathAllKeys(vector<string>& grid) {

}
};
```

Java:

```
class Solution {
public int shortestPathAllKeys(String[] grid) {

}
}
```

Python3:

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