**980. Unique Paths III**

Hard

130988Add to ListShare

On a 2-dimensional grid, there are 4 types of squares:

* 1 represents the starting square.  There is exactly one starting square.
* 2 represents the ending square.  There is exactly one ending square.
* 0 represents empty squares we can walk over.
* -1 represents obstacles that we cannot walk over.

Return the number of 4-directional walks from the starting square to the ending square, that **walk over every non-obstacle square exactly once**.

**Example 1:**

**Input:** [[1,0,0,0],[0,0,0,0],[0,0,2,-1]]

**Output:** 2

**Explanation:** We have the following two paths:

1. (0,0),(0,1),(0,2),(0,3),(1,3),(1,2),(1,1),(1,0),(2,0),(2,1),(2,2)

2. (0,0),(1,0),(2,0),(2,1),(1,1),(0,1),(0,2),(0,3),(1,3),(1,2),(2,2)

**Example 2:**

**Input:** [[1,0,0,0],[0,0,0,0],[0,0,0,2]]

**Output:** 4

**Explanation:** We have the following four paths:

1. (0,0),(0,1),(0,2),(0,3),(1,3),(1,2),(1,1),(1,0),(2,0),(2,1),(2,2),(2,3)

2. (0,0),(0,1),(1,1),(1,0),(2,0),(2,1),(2,2),(1,2),(0,2),(0,3),(1,3),(2,3)

3. (0,0),(1,0),(2,0),(2,1),(2,2),(1,2),(1,1),(0,1),(0,2),(0,3),(1,3),(2,3)

4. (0,0),(1,0),(2,0),(2,1),(1,1),(0,1),(0,2),(0,3),(1,3),(1,2),(2,2),(2,3)

**Example 3:**

**Input:** [[0,1],[2,0]]

**Output:** 0

**Explanation:**

There is no path that walks over every empty square exactly once.

Note that the starting and ending square can be anywhere in the grid.

**Note:**

1. 1 <= grid.length \* grid[0].length <= 20