Rotten Oranges

Given a grid of dimension **nxm** where each cell in the grid can have values 0, 1 or 2 which has the following meaning:
0 : Empty cell
1 : Cells have fresh oranges
2 : Cells have rotten oranges

We have to determine what is the minimum time required to rot all oranges. A rotten orange at index [i,j] can rot other fresh orange at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (**up**, **down**, **left** and **right**) in unit time.

Example 1:

```
Input: grid = {{0,1,2},{0,1,2},{2,1,1}}
Output: 1
Explanation: The grid is-
0 1 2
0 1 2
2 1 1
Oranges at positions (0,2), (1,2), (2,0)
will rot oranges at (0,1), (1,1), (2,2) and
(2,1) in unit time.
```

Example 2:

```
Input: grid = {{2,2,0,1}}
Output: -1
Explanation: The grid is-
2 2 0 1
Oranges at (0,0) and (0,1) can't rot orange at
(0,3).
```

Your Task:
You don't need to read or print anything, Your task is to complete the function **orangesRotting()** which takes grid as input parameter and returns the minimum time to rot all the fresh oranges. If not possible returns -1.

Expected Time Complexity: O(n*m)
Expected Auxiliary Space: O(n*m)

Constraints:
$1 \leq n, m \leq 500$