```
def fill_gaps(grid: list[list[int]]) -> list[list[int]]:
new_grid = copy.deepcopy(grid)
```

a.) `new_grid = copy.deepcopy(grid)`
b.) `row = 0`
c.) `col = 0`
d.) `neighbour_list = []`
e.) `for i in grid:`
f.) `    for j in i:`
g.) `        if j == 0:`
h.) `            neighbour_list = find_neighbor_values(grid,row,col)`
i.) `            avg = sum(neighbour_list) // len(neighbour_list)`
j.) `            new_grid[row][col] = avg`
k.) `        col += 1`
l.) `    row += 1`

m.) `return new_grid`

1	0
3	4

b.) row = 0
c.) col = 0
d.) taking an empty list for neightbour list
e.) i = [1,0]
f.) j = 0
g.) if j == 0 which is true this time
h.) the neighbour list looks for the surrounding numbers for j == 0 which is [1,3,4]
i.) the variable avg finds the avg of the surrounding numbers which is [1+3+4] = 8 and divides it by the number of lists which is 3. Thus 8/3=2.667≈ 3
j.) this line replaces the 0 with 3 which is the estimated value of the plot. Thus new grid is **[** [1,3], [3,4] **]**
k.) here the value of column is 1 since [**col = col + 1**]
l.) here the value of row is 1 since [**row = row + 1**]
m.) we return to the loop again
b.) row = 1
c.) col = 1
d.) taking an empty list for neightbour list
e.) i = [3,4]
f.) j = 3 which is not equals to 0
g.) if j == 0 which is false this time thus we do not go through the h,i and k
k.) col = 2
l.) row = 2
Thus we get that there are 2 columns and 2 rows in the function where we get the final matrix as

1	3
3	4

which is the new_grid.