

```

"""
Backtracking - How many spaces in the matrix are reachable
if a spot is reachable if the sum of the digits is not
greater than the given threshold

example: row 13, col 56 == 1+3+5+6 = 15
"""

def get_reachable_count(matrix, threshold):
    rows = len(matrix)
    cols = len(matrix[0])

    if rows < 1 or cols < 1 or threshold < 0:
        return 0

    visited = [[0] * cols for row in range(rows)]

    count = reachable_count_util(matrix, threshold, visited, 0, 0)

    return count

def reachable_count_util(matrix, threshold, visited, row, col):
    count = 0
    rows = len(matrix)
    cols = len(matrix[0])

    if row < 0 or row >= rows or col < 0 or col >= cols \
        or (sum_digits(row) + sum_digits(col) > threshold) \
        or visited[row][col] is True:
        return 0

    visited[row][col] = True

    count += 1
    count += reachable_count_util(matrix, threshold, visited, row - 1, col)
    count += reachable_count_util(matrix, threshold, visited, row + 1, col)
    count += reachable_count_util(matrix, threshold, visited, row, col - 1)
    count += reachable_count_util(matrix, threshold, visited, row, col + 1)

    return count

def sum_digits(num):
    return sum(map(int, list(str(num))))

matrix1 = [[0] * 5 for _ in range(5)]

assert get_reachable_count(matrix1, 4) == 15
assert get_reachable_count(matrix1, 5) == 19
assert get_reachable_count(matrix1, 6) == 22
assert get_reachable_count(matrix1, 7) == 24
assert get_reachable_count(matrix1, 8) == 25

matrix2 = [[0] * 20 for r in range(20)]

assert get_reachable_count(matrix2, 12) == 287

matrix3 = [[0] * 14 for x in range(3)]

assert get_reachable_count(matrix3, 14) == 42

matrix4 = [[0] * 1 for y in range(1)]

assert get_reachable_count(matrix4, 0) == 1

matrix5 = [[0] * 1 for z in range(29)]

assert get_reachable_count(matrix5, 10) == 29

```