

Analysis: Spiraling Into Control

Test Sets 1 and 2

You may or may not have run across the old chestnut of a technical interview question that asks you to number all of the cells of a grid in a spiral pattern. In this problem, doing that is potentially helpful for the first two test sets. However, the last test set can be (and must be) solved without explicitly creating and numbering a spiral. (For a case with $N = 9999$, we would need to store and number almost 10^8 cells!) We'll worry about that later in the analysis.

To make a numbered spiral, we can create a grid and loop through it, starting from the upper left cell, and making a 90 degree turn to the right each time we encounter a grid boundary or an already-numbered cell. One clean way to do this is to use an array of "directions": $((0, 1), (1, 0), (0, -1), (-1, 0))$, where the starting direction $(0, 1)$ means stay in the same row and move one column right, and so on, with the other three referring to moves downward, leftward, and upward, respectively, in the grid. When our current direction is $(\Delta r, \Delta c)$ and we are at cell (r, c) in the grid, we check cell $(r + \Delta r, c + \Delta c)$ to see if it is outside the grid or has already been numbered. If so, we choose the next direction in the direction vector, looping back to the start if we go off the end, and then calculate $(r + \Delta r, c + \Delta c)$ using the new $(\Delta r, \Delta c)$. Then, whether or not we changed direction, we label the current cell, increment our label counter, and proceed to the cell $(r + \Delta r, c + \Delta c)$. We stop when we label the N^2 -th cell.

For Test Set 1, we can exhaustively enumerate all legal paths, using our spiral numbering to determine which moves are allowed. We can keep track of the shortcuts taken in each possible path, and see whether any path finishes in exactly K moves. Since there is a shortcut to take (or not take) in almost every cell, this solution is exponential.

For Test Set 2, we can refine this idea to avoid explicitly considering all possible paths. For each cell in the grid, we create an array that can hold up to one path for every possible number of moves "so far". Then we proceed along the spiral in consecutive numerical order. For every cell we visit, for every path in that cell's array, we try to extend it into all legal neighboring cells.

For example, in a 5×5 grid, we start at cell 1 with only a way to get there in 0 moves. We tell each of the legal neighboring cells (2 and 16) that we have a way to get there in 1 move, starting from cell 1. Then, in cell 2, we tell each of cells 3 and 17 that we have a way to get there in 2 moves, with the starting prefix 1, 2, and so on. The key difference from the Test Set 1 solution is that when a cell is offered a path and it already has a path with exactly that number of moves, it does not store the new path. This cuts down on the proliferation of possible paths.

And so we have a [dynamic programming](#) / [memoization](#) solution. Since there is at most one shortcut to take in each cell, we do constant work per cell; since we are populating a table that is $N^2 \times N$, this solution takes $O(N^3)$ time.

Test Set 3

To be able to handle grids with N up to 9999, we need to do three things efficiently:

- Given a value of K , determine whether a path with exactly K moves exists.
- Find such a path.
- Given the coordinates of a grid cell, find its number in the spiral.

We can begin by observing that the following values of K are IMPOSSIBLE:

1. $K < N - 1$. In this case, even if we move as directly as possible toward the center (taking only shortcuts), there are not enough moves to get there.
2. Odd K . We can see this via a "checkerboard argument". Imagine that the grid has a [checkerboard pattern](#), with the upper left cell being black. Then the diagonal running from the upper left cell to the lower right cell is entirely black, and so the central cell is black. Now, each move is in an orthogonal (i.e., "compass") direction, so each move either takes us from a black cell to a red cell or vice versa. Therefore, to end on the central cell, which is black, we must make an even number of moves.

As it turns out, these are the *only* impossible cases! To see why, it helps to think of the spiral as consisting of concentric square rings, with "ring" 0 being just the central cell, ring 1 being the eight cells around that, ring 2 being the sixteen cells around ring 1, and so on. For example, the rings in the $N = 7$ grid look like this:

```

3333333
3222223
3211123
3210123
3211123
3222223
3333333

```

In ring 1, there are three possible shortcuts: we can move into ring 0 from the top, right, or bottom, and this will save us 6, 4, or 2 moves, respectively, compared to just walking through all of ring 1 (and into ring 0) without taking a shortcut. But we can only take one of these shortcuts.

In ring 2 (and beyond), for simplicity, let's only consider taking shortcuts that are in the same row or column as the central cell. There are four such shortcuts, and they will save us 14, 12, 10, or 8 moves, respectively. Notice that if we take the 8-move shortcut from ring 2, for example, we cannot use any of the shortcuts in ring 1. However, if we take the 14-move shortcut from ring 2, we will still have access to all three shortcuts in ring 1.

Similarly, in ring 3, the shortcuts save us 22, 20, 18, or 16 moves, and if we take the first of these, we still have access to all four shortcuts in ring 2.

We can turn these observations into a constructive solution for any even K that has an answer:

- If we want to save between 2 and 22 moves, we take the specific shortcut with that savings.
- If we want to save between 24 and 36 moves, we can take the 22-move shortcut and then take the specific shortcut (from ring 1 or 2) that gets us the remaining savings.
- If we want to save 38, 40, or 42 moves, we can take the 22-move and 14-move shortcuts and then the 2, 4, or 6-move shortcut from ring 1.
- We already established that we cannot save 44 moves or more (since then K would be less than $N - 1$).

Generalizing this strategy: we can find the number of moves we need to save (which is $N^2 - 1 - K$; call it s), and then proceed from the outermost ring to the innermost. At each ring r ,

1. If s is greater than or equal to the number of moves saved by the largest shortcut (i.e., $8r - 2$ moves), we take that shortcut, subtract its savings from s , and proceed to the $r - 1$ -th ring.
2. If s is equal to the number of moves saved by some other shortcut in ring r (i.e., $8r - 4$, $8r - 6$, or $8r - 8$), we take that shortcut and stop. (We must be careful not to do this when

$8r - 8 = 0$, since that move — from cell $\mathbf{N}^2 - 1$ to cell \mathbf{N}^2 — saves us nothing and is not a shortcut!

3. Otherwise, we reject all shortcuts in ring r , and proceed to ring $r - 1$.

All that remains is to find the room numbers for the shortcuts that we take. Instead of trying to generate the entire spiral, we can notice that, say, the upper left corners of the rings (starting in the central cell and going outward) have values $\mathbf{N}^2, \mathbf{N}^2 - 8, \mathbf{N}^2 - 8 - 16, \mathbf{N}^2 - 8 - 16 - 24$, etc. We can even turn this into a formula: the upper left cell of ring r has number $\mathbf{N}^2 - 8 \sum_{i=0}^r i = \mathbf{N}^2 - 4(r)(r + 1)$.

Once we have the number of the upper left cell of a ring, it is not too hard to find the values of the cells we are using for shortcuts in that ring (i.e., the ones in the central row or column). If the upper left cell's number is x , the shortcuts have numbers $x + r, x + 3r, x + 5r, x + 7r$.

Because there are $\frac{\mathbf{N}+1}{2}$ rings in total, and we do $O(1)$ work in each of them, this solution is $O(\mathbf{N})$ and easily passes Test Set 3. There are values of \mathbf{K} that require a shortcut to be used in every ring (except the central cell), so we can't do better than this in general.

It could have been worse...

We thought about presenting this problem in a different way, without mentioning spirals: starting from the upper left cell of an $\mathbf{N} \times \mathbf{N}$ grid, give a path that reaches the central cell in exactly \mathbf{K} moves, or say it is impossible. One solution is to come up with the spiral path and strategy from this problem! But there were many other time-wasting roads to go down there, and the problem was already challenging for the first problem of a Round 2.