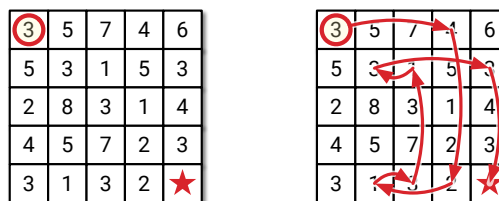


For each of the problems below, transform the input into a graph and apply a standard graph algorithm that you've seen in class. Whenever you use a standard graph algorithm, you *must* provide the following information. (I recommend actually using a bulleted list.)

- What are the vertices? What does each vertex represent?
- What are the edges? Are they directed or undirected? If it matters to your algorithm, is the graph a dag?
- If the vertices or edges have any associated values, what are they?
- **What is the precise relationship between your graph and the stated problem?**
- What standard graph algorithm are you using to solve the problem? Exactly how are you invoking that algorithm, and exactly how are you using the results?
- What is the running time of your entire algorithm, *including* the time to build the graph, as a function of the original input parameters?

Finally, remember that even when you are explicitly given a graph as part of the input, that may not be the graph you want to use as input to the standard algorithm!

A **number maze** is an $n \times n$ grid of positive integers, represented as an array $M[1..n, 1..n]$. A token starts in the upper left corner; your goal is to move the token to the lower-right corner. On each turn, you can move the token in any direction (up, down, left, or right) by a distance equal to the number on its current square. For example, if the token is on a square labeled 3, then you can move it three steps up, three steps down, three steps left, or three steps right on your next turn. However, moving the token off the edge of the board is forbidden.



A 5×5 number maze that can be solved in eight moves.

1. Describe and analyze an efficient algorithm that either returns the minimum number of moves required to solve a given number maze, or correctly reports that the maze has no solution.

For example, given the number maze shown above, your algorithm should return the integer 8, which is the length of the sequence of moves shown on the right. (This is actually the unique shortest solution.)

The remaining problems (starting on the next page) consider variants of problem 1, where a sequence of moves must satisfy some additional constraint to be considered a valid solution. The example solution shown above is *not* a valid solution for *any* of these variants. For each problem, your goal is to describe an algorithm that either returns the minimum number of moves in a *valid* solution to a given number maze, or reports correctly that no valid solution exists.

2. Suppose a sequence of moves is considered *valid* if and only if the moves alternate between horizontal and vertical. That is, a valid move sequence never has two horizontal moves in a row or two vertical moves in a row.

Describe and analyze an algorithm that either returns the minimum number of moves in a valid solution to a given number maze, or reports correctly that no valid solution exists.

3. Suppose a sequence of moves is considered *valid* if and only if its length (the number of moves) is a multiple of 5.

Describe and analyze an algorithm that either returns the minimum number of moves in a valid solution to a given number maze, or reports correctly that no valid solution exists.

Harder problems to think about later:

4. Now suppose a sequence of moves is considered *valid* if and only if it does not contain two adjacent moves in opposite directions. In other words, a sequence of moves is valid if and only if it contains no U-turns.

Describe and analyze an algorithm that either returns the minimum number of moves in a valid solution to a given number maze, or reports correctly that no valid solution exists.

5. Now suppose a sequence of moves is considered *valid* if and only if each move in the sequence is longer than the previous move (if any). In other words, a sequence of moves is valid if and only if the sequence of move *lengths* is increasing.

Describe and analyze an algorithm that either returns the minimum number of moves in a valid solution to a given number maze, or reports correctly that no valid solution exists.

6. Finally, suppose a sequence of moves is considered *valid* if and only if the sequence of move lengths is a palindrome. (A palindrome is any sequence that is equal to its reversal.)

Describe and analyze an algorithm that either returns the minimum number of moves in a valid solution to a given number maze, or reports correctly that no valid solution exists.