Problem 241: Pathfinder

Difficulty: Hard

Author: Jonathan Tran, Dallas, Texas, United States

Originally Published: Code Quest 2024

Problem Background

Congratulations! You've been hired for an internship with Lockheed Martin Code Quest! Your team is going to be working on some autonomously driven robots. Your first task is to develop a program to help the robot identify how many unique paths can be taken through a maze.

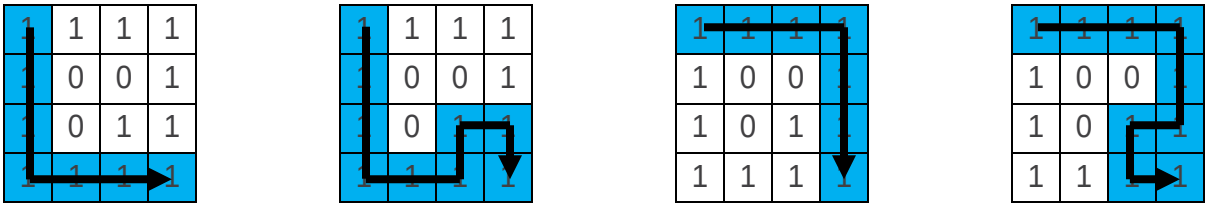
Problem Description

Positions in the square maze will either be open (indicated by a value of 1) or blocked by an obstacle (marked with a 0). The robot can only move to positions that do not contain obstacles, and can only move one step at a time in orthogonal directions (up, down, left, or right – not diagonally).

Your solution should find every unique path from the starting point to the destination cell that contains only open cells (the starting point and destination are both guaranteed to be open cells themselves, and there will always be at least one valid path). Paths may not backtrack or cross over each other (in other words, you cannot enter a cell you'd already previously visited on the same path).

In each maze, the start position will be the top left corner - (0,0) – and the destination will be the bottom right corner (N-1,N-1), where N is the size of the maze.

For example, consider the maze below. There are four unique paths through this maze, marked with the blue cells.



Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A line containing a positive integer, N, representing the size of the maze in each dimension
- N lines, each containing N binary digits (0 or 1), representing the layout of the maze

```
1
4
1111
1001
1011
1111
```

Sample Output

For each test case, your program must print a single line containing an integer representing the number of unique paths that can be taken through the given maze.

```
4
```