Approach 1: Union-Find

并查集

**Intuition**

To find the number of components in a graph, we can use either depth-first search or union find. The main difficulty with this problem is in specifying the graph.

为了寻找图的组成部分的个数，可用dfs或并查集。主要困难是如何分析这个题目的图。

One "brute force" way to specify the graph is to associate each grid square with 4 nodes (north, south, west, and east), representing 4 triangles inside the square if it were to have both slashes.

一种方式是把图中每个网格分成四个方向的节点(北、南、西、东)，若网格内有两条斜线代表有4个三角形。

Then, we can connect all 4 nodes if the grid square is " ", and connect two pairs if the grid square is "/" or "\". Finally, we can connect all neighboring nodes (for example, the east node of the square at grid[0][0] connects with the west node of the square at grid[0][1]).

然后，如果网格是空我们链接全部4个节点；如果为/或\我们链接两对。

最后，我们链接所有相邻的节点。

This is the most straightforward approach, but there are other approaches that use less nodes to represent the underlying information.

**Algorithm**

Create 4\*N\*N nodes, one for each grid square, and connect them as described above. After, we use a union find structure to find the number of connected components.

We will skip the explanation of how a DSU structure is implemented. Please refer to <https://leetcode.com/problems/redundant-connection/solution/> for a tutorial on DSU.