In this assignment you will use the disjoint set data structure to build a maze. Code is provided for all drawing operations so you have to focus on the maze-building part only. You will need to modify the maze.java program provided in the class web page. That program makes calls to the StdDraw.java library, which can be found at http://algs4.cs.princeton.edu/stdlib/StdDraw.java.html

To build a maze, you must do the following. Let \( M \) be the number of rows and \( N \) be the number of columns of your square maze. When all walls are present (see figure), each of the \( M \times N \) cells in the maze belongs to a different set. Thus you have \( M \times N \) sets in your disjoint set forest. When you remove a wall, if the cells that were separated by that wall belonged to different sets, you must unite these sets. This process is repeated until all cells belong to a single set; at that point you display the maze. This guarantees that there is a path from the start to the finish and that every cell in the maze is reachable (can you see why?).

The following pseudocode illustrates the process to build the maze:

Create full maze with four walls around each cell
Assign each cell to a different set in a disjoint set forest \( S \)
While \( S \) has more than one set
   Select a random cell \( C \)
   Select randomly one of the walls around \( C \) (north, south, east, or west)
   If the cells separated by the chosen wall belong to different sets
      unite the sets and remove the wall
   otherwise do nothing
Display maze

As usual, write a report describing your work.
Example of wall removal. The east wall of cell 2,5 has been removed from the $5 \times 5$ maze.