```java
// 8 directional movement in a grid
int dx[] = { 1, 1, 0, -1, -1, -1, 0, 1 };
int dy[] = { 0, -1, -1, -1, 0, 1, 1, 1 };
// Function to find the number of islands.
public int numIslands(char[][] grid) {
    int row = grid.length;
    int col = grid[0].length;
    boolean[][] visited = new boolean[row][col];
    int cnt = 0;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            if (grid[i][j] == '1' && !visited[i][j]) {
                dfs(i, j, row, col, grid, visited);
                cnt++;
            }
        }
    }
    return cnt;
}
public void dfs(int r, int c, int row, int col, char[][] grid,
        boolean[][] visited) {
    visited[r][c] = true;
    // try all possible movement from point(r,c)
    for (int i = 0; i < dx.length; i++) {
        int newr = r + dx[i];
        int newc = c + dy[i];
        // check the valid island point
        if (newr >= 0 && newr < row && newc >= 0 && newc < col &&
                grid[newr][newc] == '1' && !visited[newr][newc]) {
            dfs(newr, newc, row, col, grid, visited);
        }
    }
}
```