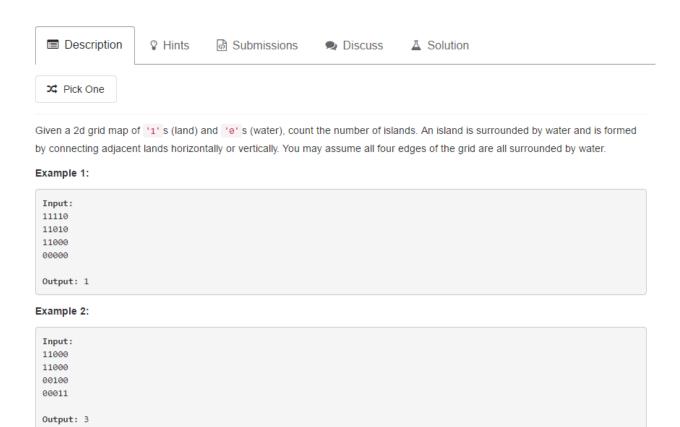
## 200. Number of Islands



```
* 这个题目是计算岛屿的数量,遇到一个1,则把周围所有的全部弄为0(这个部分要利用到深度遍历)
public class L200 {
      public int numIslands(char[][] grid) {
           if(grid == null || grid.length == 0 || grid[0].length == 0) {
           int rows = grid.length;
           int cols = grid[0].length;
           int count = 0;
           for(int i = 0; i < rows; i++)</pre>
                for(int j = 0; j < cols; j++) {</pre>
                    if (grid[i][j] == '1') {
                       count ++;
                       dfsSearch(grid, i, j, rows, cols);
                  }
                }
           return count++;
      //每次遇到一个'1'后,开始向四个方向递归搜索,搜到后变为'0'
       //因为相邻的属于一个island,然后开始继续找下一个'1',这儿就是一个深度遍历算法
      private void dfsSearch(char [][] grid, int i, int j, int rows, int cols) { if (i < 0 || i >= rows || j < 0 || j >= cols) {
             return;
           if(grid[i][j] != '1')
                return ;
           grid[i][j] = '0';
           dfsSearch(grid, i+1, j, rows, cols);
dfsSearch(grid, i-1, j, rows, cols);
dfsSearch(grid, i, j+1, rows, cols);
           dfsSearch(grid, i, j-1, rows, cols);
}
```