

数独解析

成员：

陈杭昇（17345008）

蒋青海（17345023）

陈潇（17345012）

所属班级：数学学院（珠海）2017 级 1 班

解题思路：

基本上是暴力算法，优化主要在于填格子的先后顺序确定方面。具体而言，就是每一次填的时候，都是先计算出目前可填数字最少的格子 A，从 A 格开始填。

伪代码：

1. 输入，初始化数组。
2. 当数独没有填完的时候：
 - 1) 若是第一次填，或者之前填了格子，则在填了之后的数独的基础上，计算出所对应的可填数字最少的格子，作为下一次待定填的格子 B。
 - 2) 不断用不同数字尝试填这个格子 B，若有一个数字 k 符合数独规则（即同一行，同一列，同一个九宫格中数字均不同），则将此格填为 k，并记录格子对应的行和列，以及 k 值。刷新相关数据，返回 2.1）。
 - 3) 若尝试完毕之后，没有数字符合数独规则，则回溯，得到上一个填的格子的信息（行列以及 k 值），将 k 值加 1，继续 2.2）。
3. 输出数组。

重点函数的作用：

1. **notinthisline**: 检查数独第 line 行中是否有数字 k，若有，则回传 false；否则回传 true。
2. **notinthisrow**: 检查数独第 row 列中是否有数字 k，若有，则回传 false；，否则回传 true。
3. **notinnine**: 检查数独同一个九宫格中是否有数字 k，若有，则回传 false；，否则回传 true。
4. **getlocal**: 根据目前数独的情况，确定出可填数字最少的格子，并回传对应的行和列。

实现难点：

1. `notinine` 函数中，如何确定一个九宫格：利用了整数除法的原理。
2. `getlocal` 函数中，如何确定可填数字最少：定义了三维数组，保存每个格子可填的数字有什么，最后比较每个格子可填数字的总数。
3. 如何回溯：在找到上一个所填格子 A 的信息之后，需要改变已经填写的格子数目（因为 A 此时重新看成是尚未填写的格子），同时回溯数组下标需减 1（具体可参看代码）。与此同时，由于 k 需要自加，故 for 循环中 k 不能定义为 0。（否则每次 k 会重新从 0 开始，会造成死循环，具体可参看代码）

使用的数据结构：

结构体、数组。