

CUDA_SUDOKU

1 背景

数独是一种基于逻辑运算的数字游戏。游戏规则为填充矩阵中空余的方格，要求填充的数字没有在该行、该列以及该块中出现过。

数独游戏的难易程度不是由已知数字的数量决定的，而是由空格中候选项的多少和尝试的深度决定的。

						8			3		7	9					2	
	9								4									
7		8	9					4				3	6			1		5
		4						7								2	1	7
					6		2					4						
8						2	3	9						9			5	
2				3														9
3				5			6		7									
	4			6		1	5									6		

上面的两个图中，左图有 24 个方格已经填充，右图有 23 个方格已经填充，也就是说，剩余的空格数目相差不多，那解题难度是不是也相差不多呢？事实上，右图难度比左图大很多。

左图中的数独可以连续的唯一确定某一个空格中的数字，直到所有的空格都被填充为止，根本不需要假设某个候选项是正确的，也没有所谓的尝试深度。而由图中，一开始就需要假设某个候选项是正确的，并且有较大的尝试深度，所以是困难的。

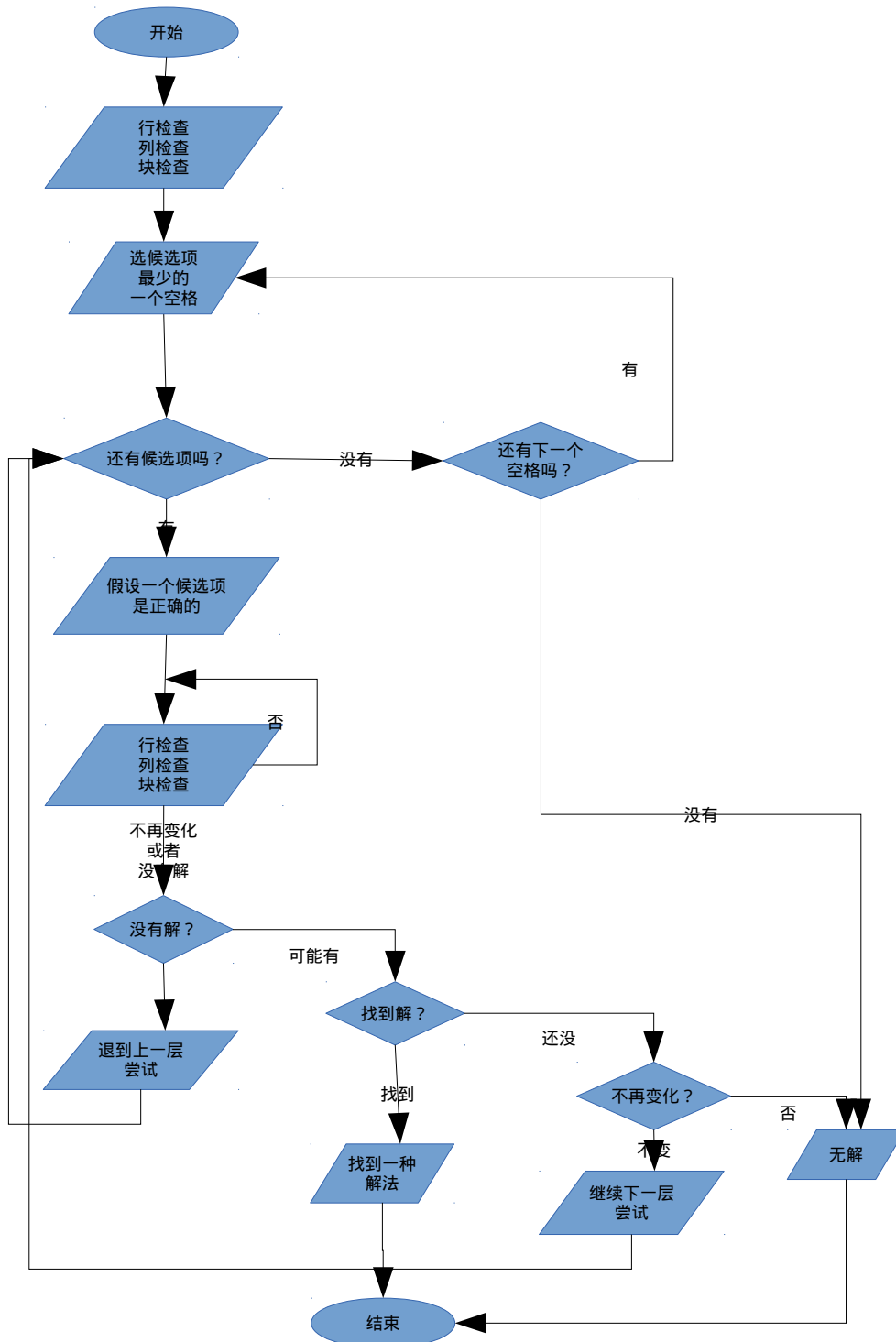
2 算法设计

- 1、每一个空位都分配一个候选项列表。
- 2、对每一行,每一列,每一块进行检查,排除掉已出现数字,直到没有候选项改变为止。
- 3、假设有最少候选项的方格的最大的候选是正确的,开始尝试。
- 4、如果发现只剩下一个候选项,就确定下来。

- 5、如果发现没有候选项了,那这个分支就走错了,回退。
- 6、直到发现所有方格都确定下来了。

3 程序实现

首先，用 C 语言来实现整个数独解题算法。



然后，在此基础上，形成一个 CUDA 版本的数独解题程序，在该版本中，行检查、列检查和块检查成为了 kernel 函数，可以在显卡中并行执行。

4 测试

经过测试，程序可以正确解出上面提到的简单数独和困难数独。

3	1	7	9	4	5	8	2	6	6	1	3	2	7	4	9	8	5
4	5	6	2	1	8	7	9	3	4	9	2	8	1	5	7	6	3
8	9	2	3	6	7	1	4	5	7	5	8	9	3	6	1	4	2
9	8	4	5	3	6	2	1	7	5	2	4	1	9	3	8	7	6
7	3	5	4	2	1	9	6	8	1	3	9	7	6	8	2	5	4
2	6	1	7	8	9	3	5	4	8	7	6	4	5	2	3	9	1
1	2	8	6	5	3	4	7	9	2	6	5	3	8	7	4	1	9
6	4	9	8	7	2	5	3	1	3	8	1	5	4	9	6	2	7
5	7	3	1	9	4	6	8	2	9	4	7	6	2	1	5	3	8

5 优化

9×9 的数独解题时有 9 个线程并行执行，和显卡每个 block 能够支持的最多 1024 线程相比太少了，远没有充分利用显卡的能力。实际测试中也发现，当数独规模只有 9×9 时，CUDA 版本的解题程序不如 C 版本的程序跑的块。

我们尝试 16×16 和 25×25 规模的数独，发现规模越大，CUDA 版本和 C 版本运行时间的差距越小，当规模超过 25×25 时，CUDA 版本已经明显快于 C 版本的程序了。

对一个 25×25 完全空的数独求解，结果如下图。

25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
20	19	18	17	16	5	4	3	2	1	25	24	23	22	21	15	14	13	12	11	10	9	8	7	6
15	14	13	12	11	25	24	23	22	21	10	9	8	7	6	5	4	3	2	1	20	19	18	17	16
10	9	8	7	6	15	14	13	12	11	5	4	3	2	1	20	19	18	17	16	25	24	23	22	21
5	4	3	2	1	10	9	8	7	6	20	19	18	17	16	25	24	23	22	21	15	14	13	12	11
24	21	25	23	22	16	15	14	13	20	8	5	12	11	19	4	3	7	10	18	1	2	6	9	17
9	6	5	4	3	2	12	21	25	19	1	23	20	18	17	14	13	15	11	24	16	10	22	8	7
16	10	7	20	17	11	3	5	1	22	4	13	14	6	2	12	8	9	25	23	19	21	24	18	15
2	1	12	14	8	24	18	9	10	4	22	7	15	16	3	19	17	6	21	20	23	13	25	11	5
18	11	19	15	13	7	6	17	23	8	24	25	21	9	10	2	16	5	1	22	14	12	20	4	3
23	25	11	10	24	12	17	7	9	18	13	6	4	5	22	16	21	19	8	3	2	15	1	14	20
7	5	6	13	4	22	21	20	14	25	3	2	1	15	23	17	10	24	18	9	11	16	12	19	8
3	22	9	18	14	13	10	19	8	24	12	17	16	21	20	11	1	4	15	2	6	5	7	25	23
1	15	20	8	19	4	2	16	6	23	14	11	10	24	9	22	12	25	5	7	3	18	17	21	13
17	12	16	21	2	3	1	11	5	15	19	18	25	8	7	13	23	20	6	14	9	22	10	24	4
22	13	24	16	18	9	23	25	21	7	17	15	2	4	14	6	11	10	20	8	12	3	5	1	19
12	23	2	6	10	19	22	1	16	3	21	8	11	25	24	9	18	17	13	5	4	7	15	20	14
11	20	14	5	25	17	13	6	24	12	9	16	19	1	18	7	15	2	3	4	22	8	21	23	10
21	17	1	19	15	8	5	4	20	2	7	10	22	3	13	24	25	14	23	12	18	6	11	16	9
8	7	4	3	9	18	11	10	15	14	6	20	5	23	12	21	22	1	16	19	24	17	2	13	25
19	16	22	11	7	1	25	24	18	5	23	3	9	10	15	8	2	21	14	17	13	20	4	6	12
14	18	21	24	20	23	8	15	4	17	11	12	6	19	25	3	5	22	9	13	7	1	16	10	2
13	2	17	9	23	6	16	12	19	10	18	21	24	20	5	1	7	11	4	15	8	25	14	3	22
6	8	15	25	12	14	7	22	3	9	2	1	17	13	4	23	20	16	24	10	21	11	19	5	18
4	3	10	1	5	21	20	2	11	13	16	22	7	14	8	18	6	12	19	25	17	23	9	15	24

6 限制

数独解题器其实并不非常适合使用 CUDA 来优化，因为数独程序的控制流程比较多，这些控制流程不能和 kernel 函数很好的并行执行。而且数独程序非常适合用递归来实现，而 kernel 函数是不能使用递归的。

7 总结

CUDA 是新兴的通用 GPU 的并行编程和计算平台，它充分利用了 GPU 擅长并行计算的能力，为科学计算等数据密集型领域提供了极大便利。通过这一学期的学习和实践，我基本掌握了 CUDA 编程的知识和技能，并具备了开发实际项目的能力。