人工智能基础第二次编程

自02 彭程 2020011075

注：本次选择的是第1题数独游戏

零、数据维护

使用了一个Sudoku类，实现功能有：从文件中读取初始状态，格式化打印状态，给出每个空格的可能状态，给出可能状态数最少的空格和可能状态，填充数字到某一格，判断是否结束。

使用了一个Node类，维护搜索树，功能包括生成子节点，返回搜索到某节点的路径。

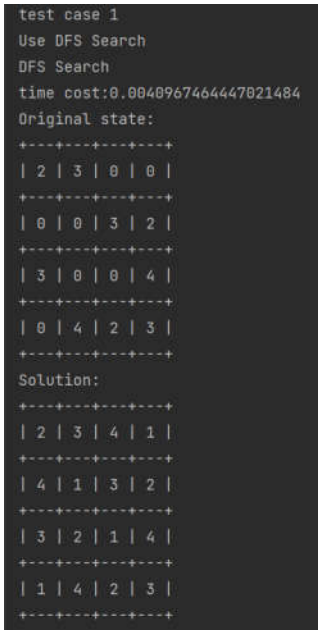
还使用若干基础类例如Stack，Queue等。

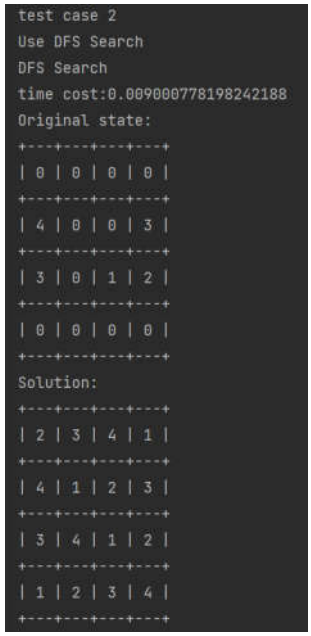
一、深度优先搜索

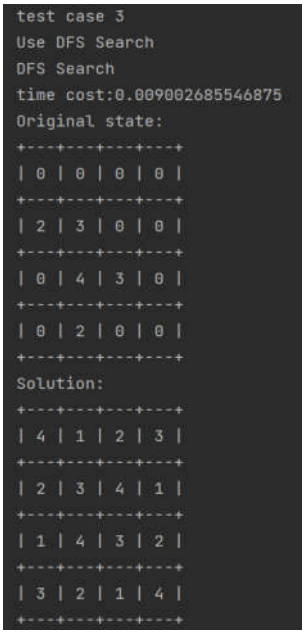
根据我的观察，游戏的复杂度应该并不高，因此我首先尝试使用DFS寻找一种可行解，主要逻辑部分的代码如下：

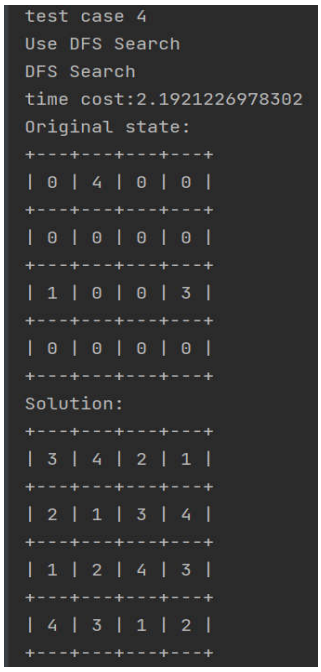
```
def DFS(problem):
    print("DFS Search")
    # 思路，深度优先搜索，适合用于搜一种解，不适合搜索全部解
    qu = Stack()
    open = Set()
    # 维护一些初始化的信息
    sudoku = problem
    node_start = Node(sudoku.matrix)
    qu.push(node_start)
    open.add(np.array(node_start.state).tobytes())
    # 开始深度优先搜索
    while not qu.empty():
        node = qu.pop()
        state = node.state
        actions = sudoku.actions(state)
        for action in actions:
            new_node = node.child_node(sudoku, action)
            bytestate = np.array(new_node.state).tobytes()

            if sudoku.is_goal(new_node.state):
                return new_node
            elif not open.find(bytestate):
                # print(new_node.state)
                qu.push(new_node)
                open.add(bytestate)
    print("no way")
    return 0
```

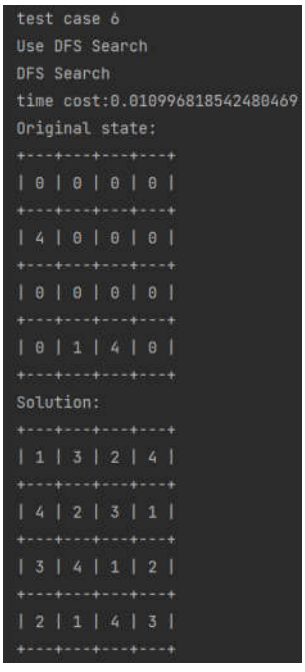
对于需要求解的六种情况，程序都可快速给出一种可行解：











二、回溯搜索

对于此类问题，我们知道回溯搜索是一种高效的解决办法，于是对于本题我还实现了回溯搜索算法，与DFS不同，回溯搜索可以高效得求得所有可能解，主要逻辑部分代码如下：

```python
def BackTracking(problem):
    # 每一步查找都加入约束，找可能值最少的点进行推断
    # 每个空位可能的action数量记录一下，然后递归推断
    # 可以用于搜索所有解
    sudoku = problem
    pos,num_list = sudoku.solve(sudoku.matrix)
    solution = []
    if len(num_list)==0:
        return []
    for num in num_list:
        new_state = deepcopy(sudoku.matrix)
        new_state[pos[0],pos[1]] = num

        if sudoku.is_goal(new_state):
            solution.append(new_state)
        else:
            new_problem = Sudoku(new_state)
            solution += BackTracking(new_problem)
    return solution
```

在回溯搜索算法中，每一步我都选择可能取值最少的点进行递归搜索，搜索结果如下图展示，由于解情况较多，此处我仅展示时间和可行解数量，依次有1、1、2、3、3、6种解。

```
test case 1
Use BackTracking Search
time cost:0.0030384063720703125
solution number:1

test case 2
Use BackTracking Search
time cost:0.0062255859375
solution number:1

test case 3
Use BackTracking Search
time cost:0.006999492645263672
solution number:2

test case 4
Use BackTracking Search
time cost:0.011979103088378906
solution number:3

test case 5
Use BackTracking Search
time cost:0.009999752044677734
solution number:3

test case 6
Use BackTracking Search
time cost:0.021384239196777344
solution number:6
```

由于要求展示搜索结果，故之后均为使用回溯搜索打印出的所有可行结果：

test case 1

Use BackTracking Search

Original state:

```
+---+---+---+---+
| 2 | 3 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 3 | 2 |
+---+---+---+---+
| 3 | 0 | 0 | 4 |
+---+---+---+---+
| 0 | 4 | 2 | 3 |
+---+---+---+---+
```

time cost:0.0040051937103271484

solution number:1

solution 1:

```
+---+---+---+---+
| 2 | 3 | 4 | 1 |
+---+---+---+---+
| 4 | 1 | 3 | 2 |
+---+---+---+---+
| 3 | 2 | 1 | 4 |
+---+---+---+---+
| 1 | 4 | 2 | 3 |
+---+---+---+---+
```

test case 2

Use BackTracking Search

Original state:

```
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 4 | 0 | 0 | 3 |
+---+---+---+---+
| 3 | 0 | 1 | 2 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
```

time cost:0.008991479873657227

solution number:1

solution 1:

```
+---+---+---+---+
| 2 | 3 | 4 | 1 |
+---+---+---+---+
| 4 | 1 | 2 | 3 |
+---+---+---+---+
| 3 | 4 | 1 | 2 |
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
```

test case 3

Use BackTracking Search

Original state:

```
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 2 | 3 | 0 | 0 |
+---+---+---+---+
| 0 | 4 | 3 | 0 |
+---+---+---+---+
| 0 | 2 | 0 | 0 |
+---+---+---+---+
```

time cost:0.011000871658325195

solution number:2

solution 1:

```
+---+---+---+---+
| 4 | 1 | 2 | 3 |
+---+---+---+---+
| 2 | 3 | 1 | 4 |
+---+---+---+---+
| 1 | 4 | 3 | 2 |
+---+---+---+---+
| 3 | 2 | 4 | 1 |
+---+---+---+---+
```

solution 2:

```
+---+---+---+---+
| 4 | 1 | 2 | 3 |
+---+---+---+---+
| 2 | 3 | 4 | 1 |
+---+---+---+---+
```

```
| 1 | 4 | 3 | 2 |
+---+---+---+---+
| 3 | 2 | 1 | 4 |
+---+---+---+---+
test case 4
Use BackTracking Search
Original state:
+---+---+---+---+
| 0 | 4 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 1 | 0 | 0 | 3 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
time cost:0.016988039016723633
solution number:3
solution 1:
+---+---+---+---+
| 2 | 4 | 3 | 1 |
+---+---+---+---+
| 3 | 1 | 2 | 4 |
+---+---+---+---+
| 1 | 2 | 4 | 3 |
+---+---+---+---+
| 4 | 3 | 1 | 2 |
+---+---+---+---+
solution 2:
+---+---+---+---+
| 3 | 4 | 1 | 2 |
+---+---+---+---+
| 2 | 1 | 3 | 4 |
+---+---+---+---+
| 1 | 2 | 4 | 3 |
+---+---+---+---+
| 4 | 3 | 2 | 1 |
+---+---+---+---+
solution 3:
```

```
+---+---+---+---+
| 3 | 4 | 2 | 1 |
+---+---+---+---+
| 2 | 1 | 3 | 4 |
+---+---+---+---+
| 1 | 2 | 4 | 3 |
+---+---+---+---+
| 4 | 3 | 1 | 2 |
+---+---+---+---+
```

test case 5

Use BackTracking Search

Original state:

```
+---+---+---+---+
| 0 | 0 | 1 | 0 |
+---+---+---+---+
| 0 | 1 | 0 | 2 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 3 |
+---+---+---+---+
```

time cost:0.01729607582092285

solution number:3

solution 1:

```
+---+---+---+---+
| 2 | 3 | 1 | 4 |
+---+---+---+---+
| 4 | 1 | 3 | 2 |
+---+---+---+---+
| 3 | 2 | 4 | 1 |
+---+---+---+---+
| 1 | 4 | 2 | 3 |
+---+---+---+---+
```

solution 2:

```
+---+---+---+---+
| 2 | 3 | 1 | 4 |
+---+---+---+---+
| 4 | 1 | 3 | 2 |
+---+---+---+---+
```

```
| 3 | 4 | 2 | 1 |
+---+---+---+---+
| 1 | 2 | 4 | 3 |
+---+---+---+---+
solution 3:
+---+---+---+---+
| 3 | 2 | 1 | 4 |
+---+---+---+---+
| 4 | 1 | 3 | 2 |
+---+---+---+---+
| 2 | 3 | 4 | 1 |
+---+---+---+---+
| 1 | 4 | 2 | 3 |
+---+---+---+---+
test case 6
Use BackTracking Search
Original state:
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 4 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 0 | 0 | 0 |
+---+---+---+---+
| 0 | 1 | 4 | 0 |
+---+---+---+---+
time cost:0.03113555908203125
solution number:6
solution 1:
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 4 | 3 | 1 | 2 |
+---+---+---+---+
| 3 | 4 | 2 | 1 |
+---+---+---+---+
| 2 | 1 | 4 | 3 |
+---+---+---+---+
solution 2:
```

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 4 | 3 | 2 | 1 |
+---+---+---+---+
| 2 | 4 | 1 | 3 |
+---+---+---+---+
| 3 | 1 | 4 | 2 |
+---+---+---+---+
```

solution 3:

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 4 | 3 | 2 | 1 |
+---+---+---+---+
| 3 | 4 | 1 | 2 |
+---+---+---+---+
| 2 | 1 | 4 | 3 |
+---+---+---+---+
```

solution 4:

```
+---+---+---+---+
| 1 | 3 | 2 | 4 |
+---+---+---+---+
| 4 | 2 | 1 | 3 |
+---+---+---+---+
| 2 | 4 | 3 | 1 |
+---+---+---+---+
| 3 | 1 | 4 | 2 |
+---+---+---+---+
```

solution 5:

```
+---+---+---+---+
| 1 | 3 | 2 | 4 |
+---+---+---+---+
| 4 | 2 | 3 | 1 |
+---+---+---+---+
| 2 | 4 | 1 | 3 |
+---+---+---+---+
| 3 | 1 | 4 | 2 |
+---+---+---+---+
```

solution 6:

```
+---+---+---+---+
| 1 | 3 | 2 | 4 |
+---+---+---+---+
| 4 | 2 | 3 | 1 |
+---+---+---+---+
| 3 | 4 | 1 | 2 |
+---+---+---+---+
| 2 | 1 | 4 | 3 |
+---+---+---+---+

Process finished with exit code 0
```