数独（前編）

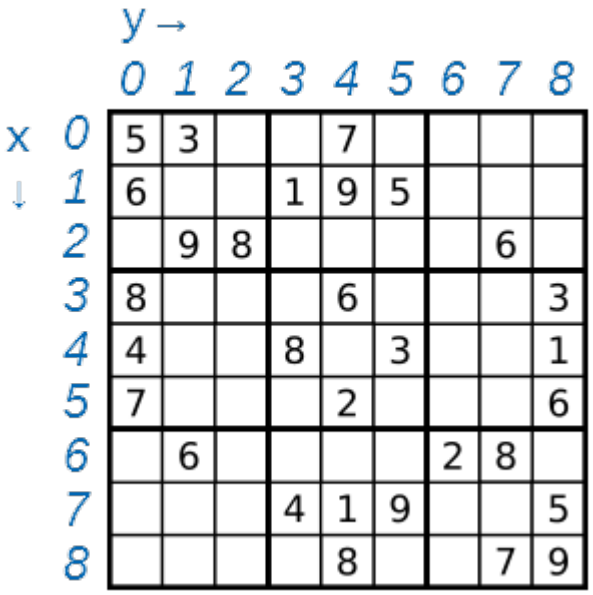
問題

ミニプロジェクト 5 と 6 では数独の問題を解くプログラムを段階的に作り上げていく（数独パズルのルールを知らない人はウィキペディア等で調べること）。ソルバー部分は第 12 回の講義で説明した深さ優先探索のプログラムをそのまま使うので、例によって `SearchProblem` クラスのサブクラスを適切に定義することが最終的な目標となるが、それはミニプロジェクト 6 で行う。ミニプロジェクト 5 ではその前段階として、数独の問題およびそれを解く過程で生ずる 9×9 のマス目における数字の配置を表す `Board` クラスを定義し、`filled`、`verify`、`get_allowed_digits`、`move` の 4 つのメソッドを実装する。[10 点]

仕様

`Board` クラスのオブジェクトは

```
board = Board([[5, 3, 0, 0, 7, 0, 0, 0, 0],
               [6, 0, 0, 1, 9, 5, 0, 0, 0],
               [0, 9, 8, 0, 0, 0, 0, 6, 0],
               [8, 0, 0, 0, 6, 0, 0, 0, 3],
               [4, 0, 0, 8, 0, 3, 0, 0, 1],
               [7, 0, 0, 0, 2, 0, 0, 0, 6],
               [0, 6, 0, 0, 0, 0, 2, 8, 0],
               [0, 0, 0, 4, 1, 9, 0, 0, 5],
               [0, 0, 0, 0, 8, 0, 0, 7, 9]])
```



のように 2 次元リストを渡されて作られる。右上図はウィキペディアにあった数独の問題だが、変数 `board` はこの問題に対応するオブジェクトとなっている。ただし数字が埋まっていないマスは 0 で表わすこととする。コンストラクタは `board.py` に与えられているように、

```
class Board:
    def __init__(self, data):
        self.data = data
```

と引数をそのままインスタンス変数 `self.data` に代入する。上図に示したように、本問では行番号を x、列番号を y とする座標系でマスの位置を表す。たとえば `self.data[x][y]`は x 行 y 列目の要素に対応する。また、コンストラクタと同様に `board.py` で与えられている`__str__`メソッドによって、`print(board)`を実行すると次ページにあるような出力が得られるようになっている（この仕組みについては教科書 276 ページ「Chapter 6-4 クラスの特殊メソッドについて」を参照のこと）。

```
+---+---+---+
|53 | 7 |   |
|6  |195|   |
| 98|   | 6 |
+---+---+---+
|8  | 6 |  3|
|4  |8 3|  1|
|7  | 2 |  6|
+---+---+---+
| 6 |   |28 |
|   |419|  5|
|   | 8 | 79|
+---+---+---+
```

実装すべきメソッド:

- `filled(self)` すべてのマスが埋まっているかどうかを `True` または `False` で返す
- `verify(self)` 数字の配置が数独のルールに違反していなかったら `True`、違反していたら `False` を返す
- `get_allowed_digits(self, x, y)` 座標（x, y）のマスに数独のルールに従って書くことが許される数字のリストを返す。数字の順序は任意。例えば `board.get_allowed_digits(6, 0)`は`[1, 3, 9]`を返す。
- `move(self, x, y, d)` 座標（x, y）のマスに数字 d を置くことによってできる新たな `Board` のインスタンスを返す。その際に、元の `Board` インスタンスの `self.data` が改変されないよう注意すること（リストのコピーについては https://goo.gl/wXtHn4 を参照のこと）。

注意

- 2019 年 1 月 18 日（金）深夜までに `board.py` を e シラバスにアップロードすること。
- 今回も単体テストを提供しているので、すべてのテストにパスすることを目指してプログラミングするとよい。ただし、採点にあたっては `board_test.py` には無いテストケースによるチェックも行われる