

# EightQueensしてみました

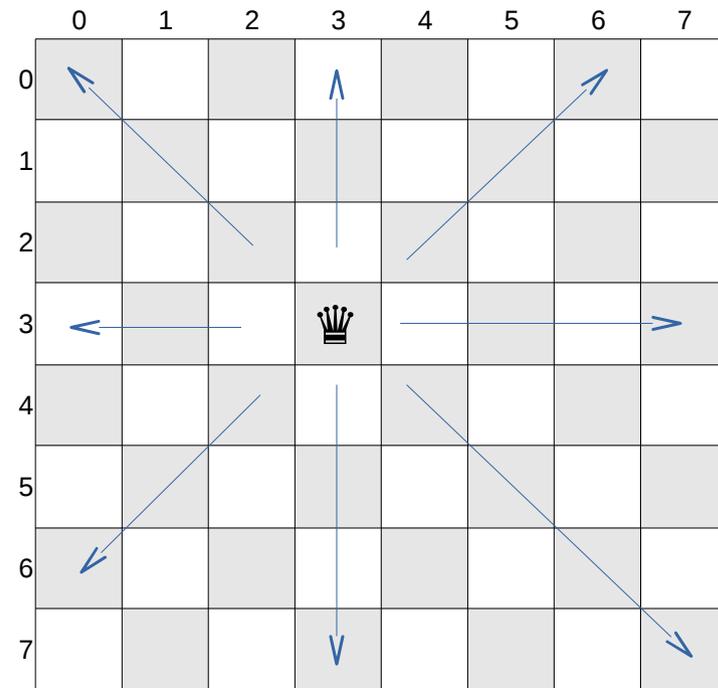
2015/07/04

東海道らぐ 2015年7月オフな集まり in 名古屋  
さとう



# まず、「えいとくいーん」について

- 古くからあるボードゲームのクイズです。
- チェスの「クイーン」8つがチェス盤に。
- クイーン同士が、互いに効いてないように並べます。
- 初心者向けのプログラム課題によく使われる問題のようです。
- 蛇足：クイーン(♔)は縦横斜めに、目一杯動けます。



# おもいきってやってみました

- 後先見ずのやっつけですが...

<https://github.com/sato-makoto/8queens>

	0	1	2	3	4	5	6	7
0				♔				
1		♔						
2								♔
3								
4	♔							
5			♔					
6					♔			
7							♔	

# で、実行結果はというと

- はい、あまり面白くないです。
- 縦横の位置を表示するだけです。
- ```
$ python eightqueen.py
```

```
[[4, 0], [1, 1], [7, 2], [0, 3], [3, 4],  
[6, 5], [2, 6], [5, 7]]
```

# やっつけな方法でやりました(0)

- はじめに、64個のマス目ぜんぶをリストアップ

```
def makemysel(myset):  
    for f1 in range(8):  
        for f2 in range(8):  
            pair = '{}-{}'.format(f1, f2)  
            myset.add(pair)  
    return myset
```

# やっつけな方法でやりました(0)

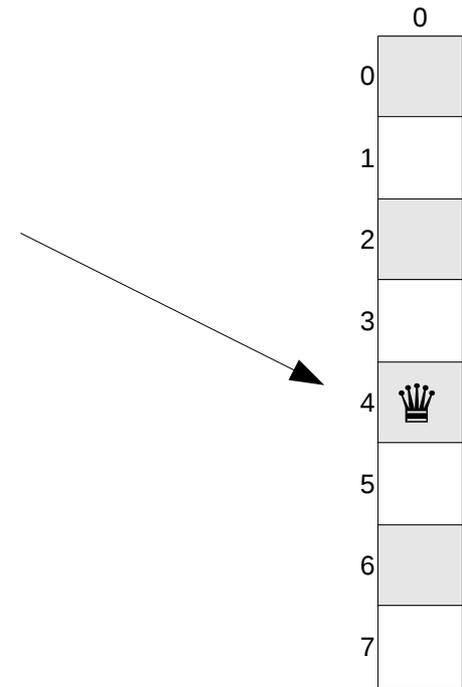
- 64個くらいなら、**たぶん**、そう大きくないですね...

```
set(['2-2', '5-4', '2-3', '7-7', '2-0', '7-6', '2-1', '7-5', '6-7', '2-6', '7-4', '2-7', '5-3', '2-4', '2-5', '5-2', '6-1', '0-4', '0-5', '0-6', '0-7', '0-0', '0-1', '0-2', '0-3', '1-1', '1-0', '1-3', '1-2', '1-5', '1-4', '1-7', '1-6', '5-6', '5-5', '6-6', '5-1', '6-4', '6-5', '6-2', '6-3', '6-0', '5-0', '4-0', '4-1', '4-2', '4-3', '4-4', '4-5', '4-6', '4-7', '3-7', '3-6', '3-5', '3-4', '3-3', '3-2', '3-1', '3-0', '7-3', '5-7', '7-2', '7-0', '7-1'])
```

# やっつけな方法でやりました(2)

- 次に、左端のマス目から、**どれか**を選びました

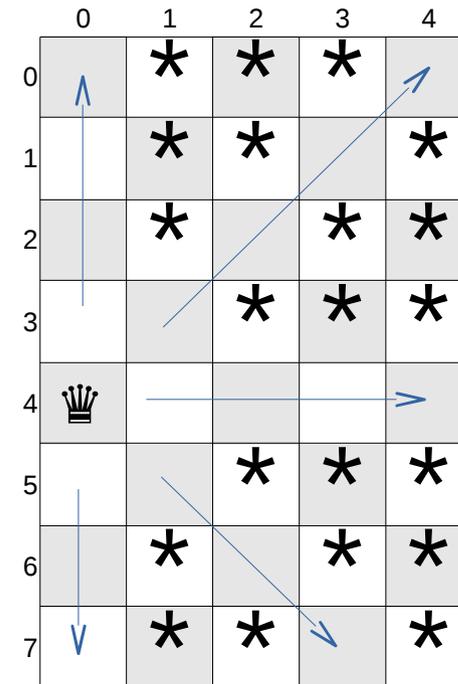
```
def randselect(n):  
    random.seed()  
    return random.randrange(0,n)
```



# やっつけな方法でやりました(3)

- そして、それより右にある、効いているマス目をぜんぶ消しました。

```
def delpair(x, y, setz):  
    pair = '{}-{}'.format(x, y)  
    if pair in setz:  
        setz.remove(pair)
```



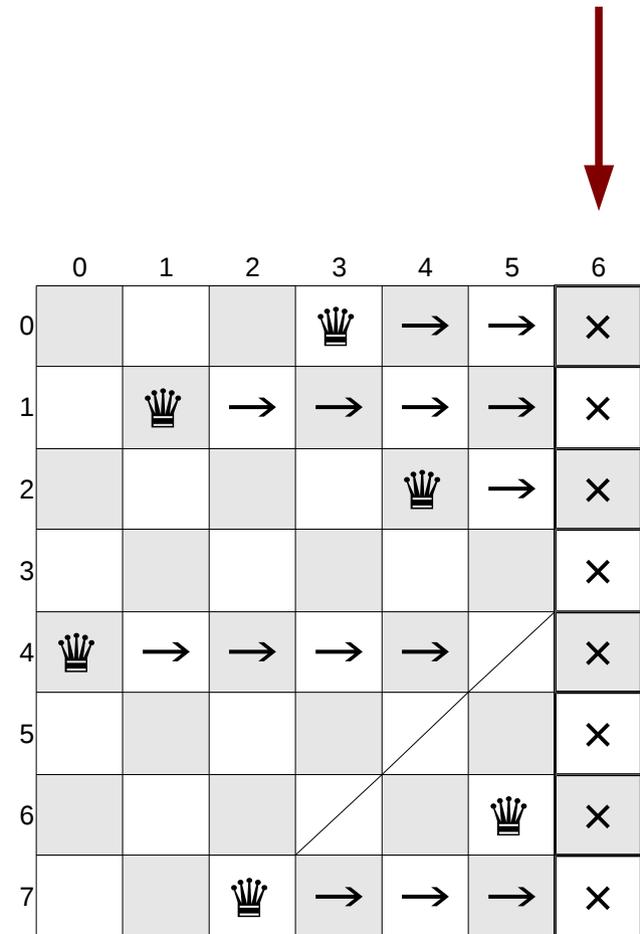
# やっつけな方法でやりました(4)

- それから、隣の行の、**まだ消えてない**マス目のどれかを選択です。

|   | 0 | 1 |
|---|---|---|
| 0 |   | * |
| 1 |   | * |
| 2 |   | ♔ |
| 3 |   |   |
| 4 | ♔ |   |
| 5 |   |   |
| 6 |   | * |
| 7 |   | * |

# やっつけな方法でやりました(4)'

- 以下、ほとんどおなじ。
- いや、もし行き詰まったら？



A 7x7 grid representing a chessboard. The columns are labeled 0 to 6 at the top. The rows are labeled 0 to 7 on the left. The grid contains a king piece (crown icon) and arrows (→) indicating movement paths. A red arrow points to column 6. The grid is as follows:

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 |   |   |   | ♔ | → | → | × |
| 1 |   | ♔ | → | → | → | → | × |
| 2 |   |   |   |   | ♔ | → | × |
| 3 |   |   |   |   |   |   | × |
| 4 | ♔ | → | → | → | → |   | × |
| 5 |   |   |   |   |   |   | × |
| 6 |   |   |   |   |   | ♔ | × |
| 7 |   |   | ♔ | → | → | → | × |

# やっつけな方法でやりました(4)"

- もし行き詰まったら？

はい、**最初から**やり直しです。

else:

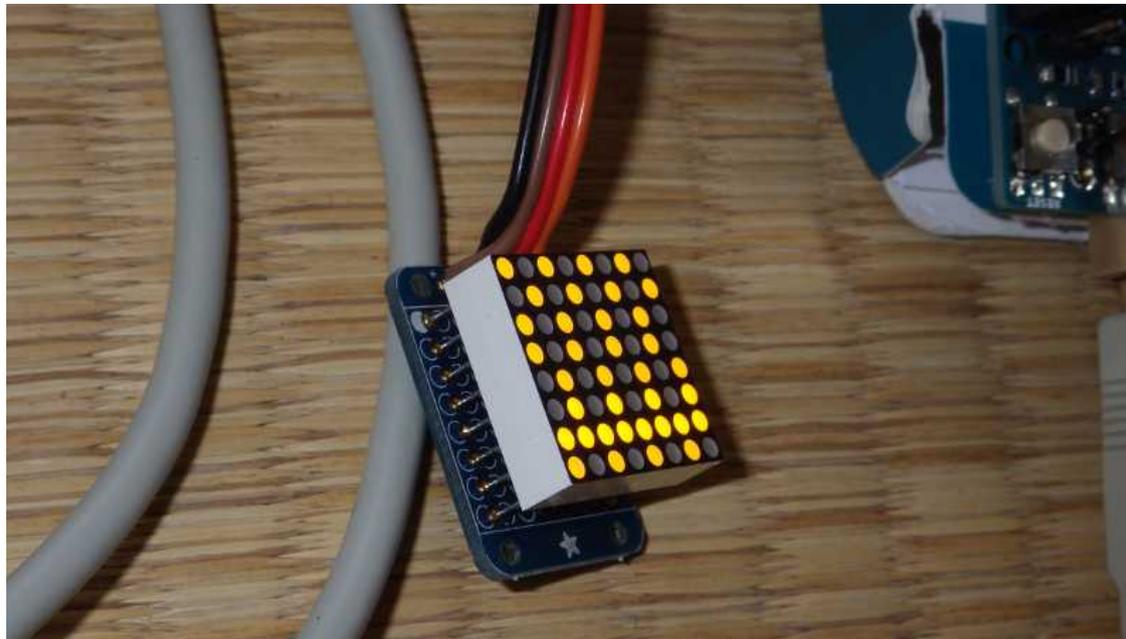
```
x, y, pairs, myset = eq.allclean()
```

- 時間がなかった(かつめんどくさい)ので、いずれは通ることを**信じて**、何度でもやり直しさせています

# もう少しおもしろくできないかと

- なんとなく買っておいた8x8のLED

<http://www.adafruit.com/products/870>



# 8x8 matrix ってこんなもの

- I2Cで制御します
- ArduinoおよびRaspberryPIで動きます
- ライブラリが公開されています(サンプルソースも)

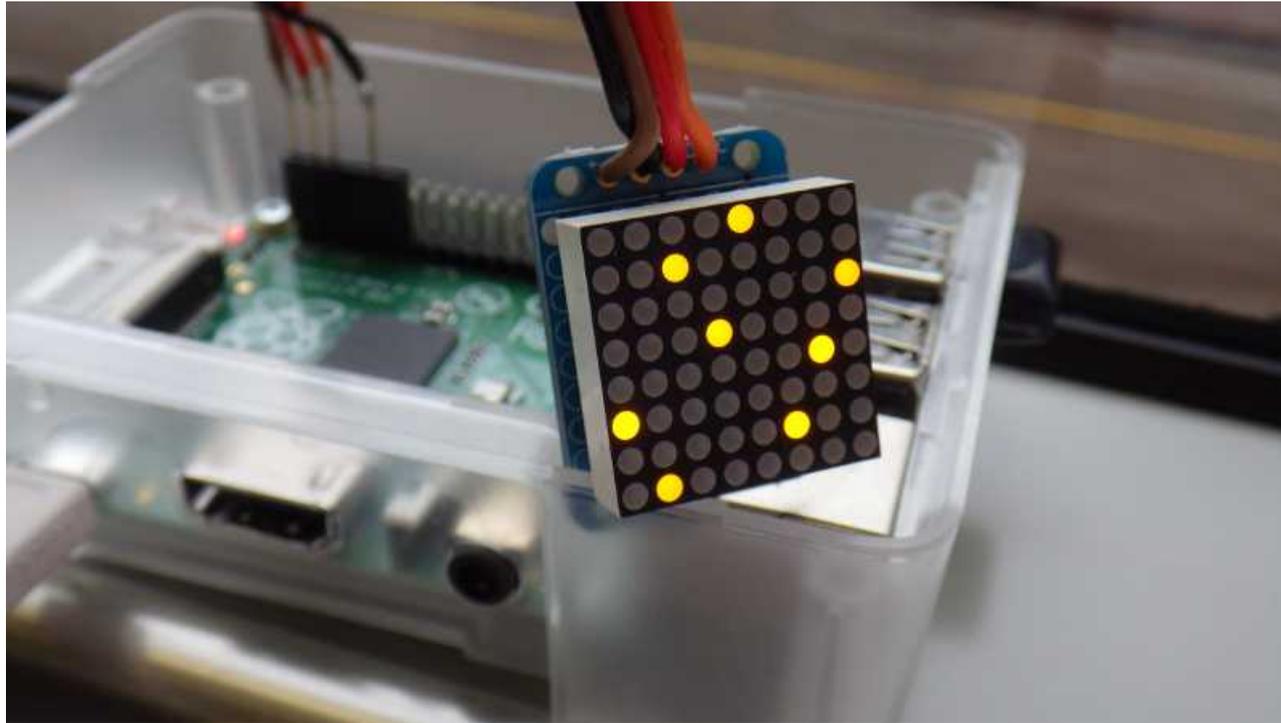
<https://github.com/adafruit/Adafruit-LED-Backpack-Library>

[https://github.com/adafruit/Adafruit\\_Python\\_LED\\_Backpack](https://github.com/adafruit/Adafruit_Python_LED_Backpack)

- **ので、プロトコルのことを何も知らなくても、ふつーに動きました**

# EightQueensが動いてるところ

- こんな感じです(じつはここに来る電車の中...)



# 実はまだ...

- LEDを、ちゃんと制御できないでいます。  
計算をスクリプトにつなげることができず、結果をコピーしています(汗

```
queen0 = [[4, 0], [1, 1], [3, 2], [5, 3], [7, 4], [2, 5], [0, 6], [6, 7]]
```

```
queen1 = [[5, 0], [3, 1], [6, 2], [0, 3], [2, 4], [4, 5], [1, 6], [7, 7]]
```

.

.

- 次回はもっとちゃんと作る、、、かもしれません

# 何かツッコミはございませんか

- クリックしてテキストの挿入

# それでは、まいど

- ご静聴だかご清澄だか、

ありがとうございます。

終