

# 畳敷き詰めマシンと 最小畳配置問題

---

fermium100

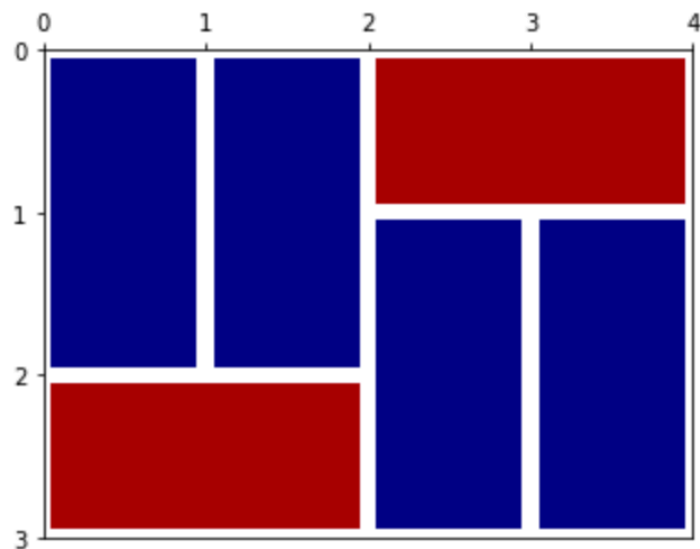
# 開発した背景

---

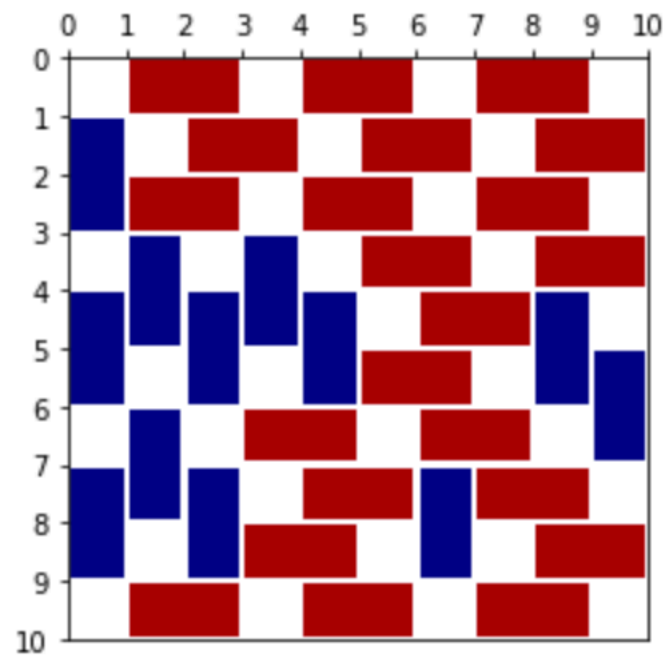
- 友人が「最小畳配置問題」について考えていた  
(出来るだけ少ない枚数の畳で、それ以上畳を置けない状態にする)
- 畳を置くか置かないかというのは二値なので、量子アニーリングが得意なのは？
- ちょうどハッカソンが開催されていたので開発してみた

# 今回開発したアプリ

## 畳敷き詰めマシン



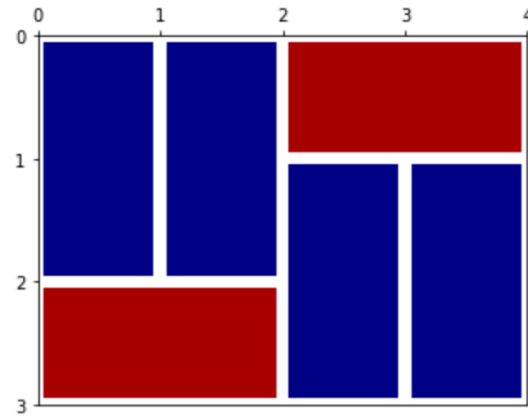
## 最小畳配置問題ソルバー



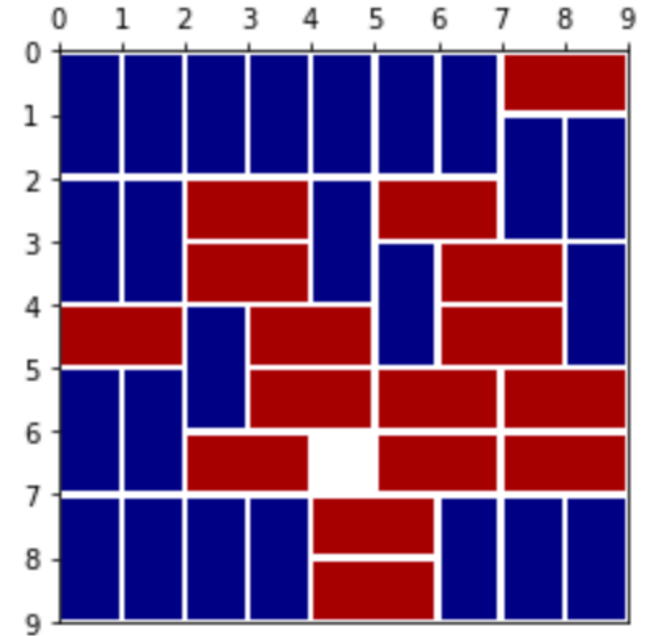
# 畳敷き詰めマシン

- 長方形の領域を畳で敷き詰める際の配置を返す

- 縦3、横4の例：



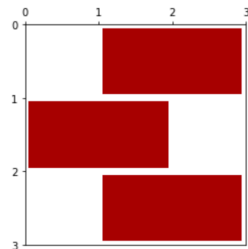
- 縦横ともに奇数のときは1マス空く (例：右図)
- 実装には`amplify.constraint.less_equal`を用いた
  - 各マスに乗る畳は1枚以下



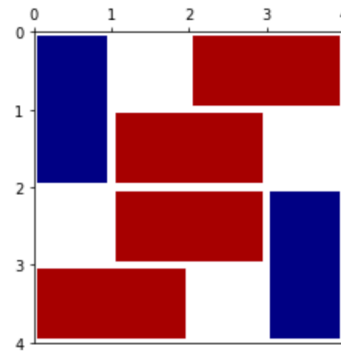
# 最小畳配置問題

- 出来るだけ少ない枚数の畳を使って、正方形(長方形)の領域をそれ以上畳が敷けない状態にする

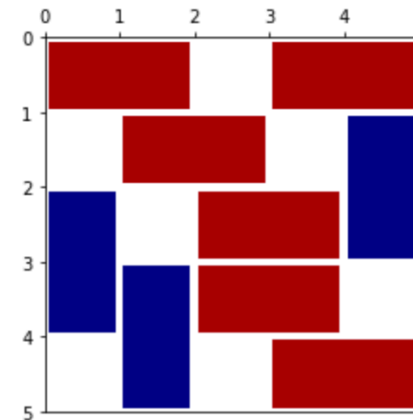
- 例：



3枚



6枚



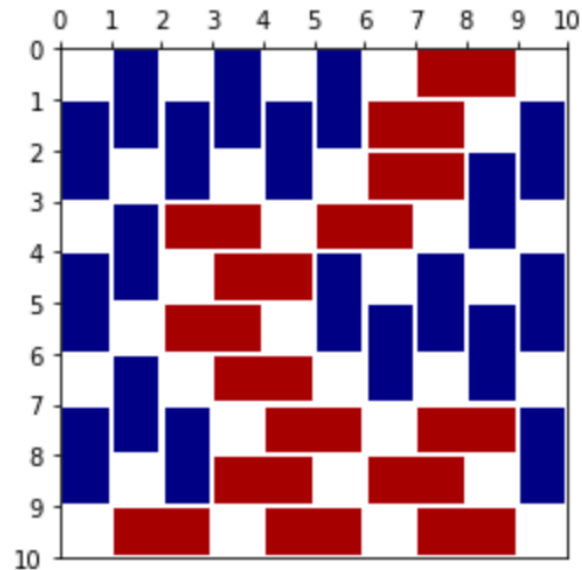
9枚

- 一般の $n$ について、 $n \times n$ の正方形で必要な畳の枚数の最小値が判明しているのは $n=3k$  ( $k$ は自然数) のときのみ

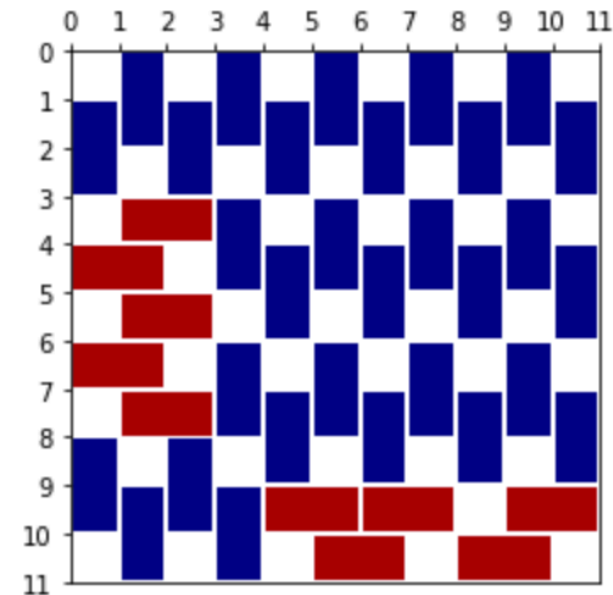
# 最小畳配置問題ソルバー

- 長方形の領域について、出来るだけ少ない枚数を使ってそれ以上畳が置けないような配置と使った畳の枚数を返す
- 10×10の実行結果
- 11×11の実行結果

畳の枚数 34



畳の枚数 41



# 最小畳配置問題ソルバー（実行結果）

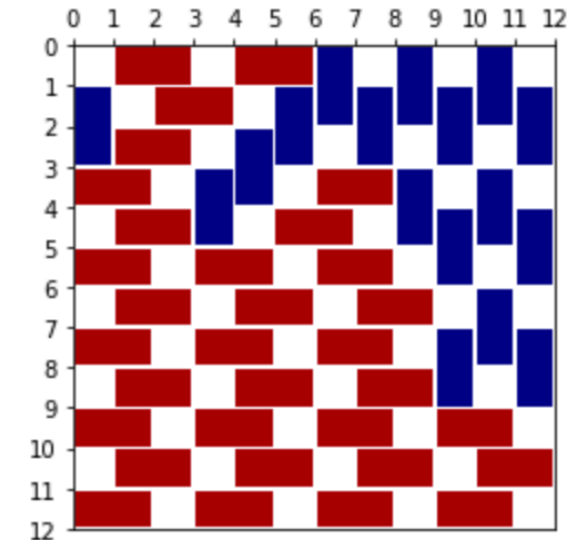
- $n$ が小さいときの最適な配置の枚数はOEIS（オンライン整数列大辞典）に載っている
- <https://oeis.org/A280984>
- $12 \times 12$ の実行結果

A280984	Minimum number of dominoes on an $n \times n$ chessboard needed to prevent placement of another domino.
---------	---

0, 2, 3, 6, 9, 12, 17, 22, 27, 34, 41, 48, 57, 66, 75, 86, 97, 108, 122, 134, 147, 163, 178, 192, 210, 227, 243, 263, 282, 300, 322, 343, 363 ([list](#); [graph](#); [refs](#); [listen](#); [history](#); [text](#); [internal format](#))

- $12 \times 12$ の最適解は48枚だが、今回のセッティング（計算時間5秒）では49枚以上の解しか得られなかった（右図）

畳の枚数 49



# 最小畳配置問題ソルバー（実装）

---

- 当初は実装に`amplify.constraint.less_equal`を用いていたが、今回のような最適化問題を解く上ではあまり良い解が得られなかった
- 代わりに、自分の手で式から考えて実装することで余計なバイアスが入らないように気をつけた
  - 畳が重ならないための制約：  
あるマス上の畳の枚数を $x$ とすると、コスト関数は  $x(1-x)$
  - 空きマスが隣り合わないための制約：  
ある隣り合う2マス上の畳の枚数を $y$ とすると、コスト関数は  $(1-y)(2-y)$