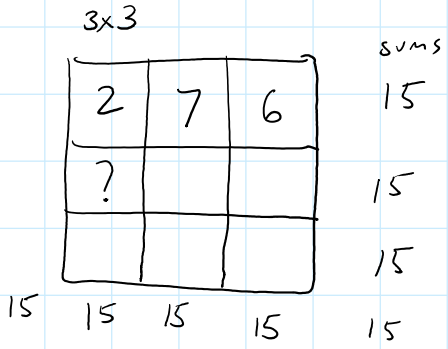
MAGIC SQUARES

$n \times n$ square, filled with $1, 2, 3, \ldots, n^2$ such that all rows, colums, and diagonals have the same sum



PROCEDURE:

States are configurations of $1, \ldots, 9$ in a 3×3 square.

Transition from one state to another by:

- swap two numbers
- swap two rows or colums

Frequency function: big for a magic square
small for a square that is far from magic

$$f(\text{square}) = \max \left| \text{sums} - 15 \right|$$

frequency distribution: $e^{-f(\text{square})/\sigma^2}$

Simulate, decreasing σ^2 slowly,
If we are lucky, the simulation will produce a magic square.

IMPLENTATION:

How do we store a squae in memory?

How do we compute row/col/diag sums?

How do we propose a random transition?

SWAPPING: How do we swap the values in two variables a and b?

~~a = b~~
~~b = a~~

```
temp = a
a = b
b = temp
```

In Python:

```
a, b = b, a
```

Replace a and b by matrix entries/rows/cols.

```
temp = np.copy(a)
a = b
b = temp
```