

# Magic Squares: function to minimize

Last time, we discussed several ideas:

Compute the absolute value differences between the row/col/diag sums with the target value.

Then: take the sum of the differences, or the average, or the max difference.

Example:

5	1	9	15
7	6	2	15
3	8	4	15
18	15	15	15

You can do swaps and get a magic square such that the max difference function monotonically decreases to zero, but the sum or average difference will increase then decrease.

```

for i in range(10000):
    ... doMove(...)

```

max steps  
↓

```

if f(state) == 0:

```

```

    break ← causes Python to quit the loop

```

# at end of loop, the value of i is the number of steps made

## Traveling Salesperson Problem (TSP)

N=20 points

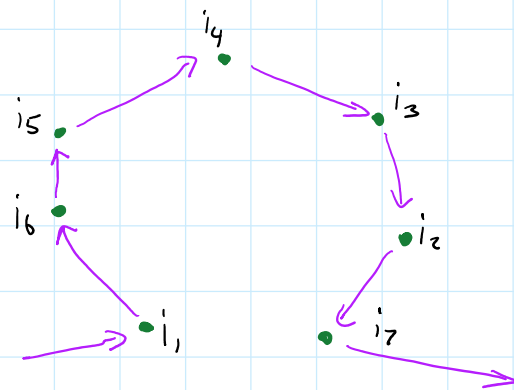
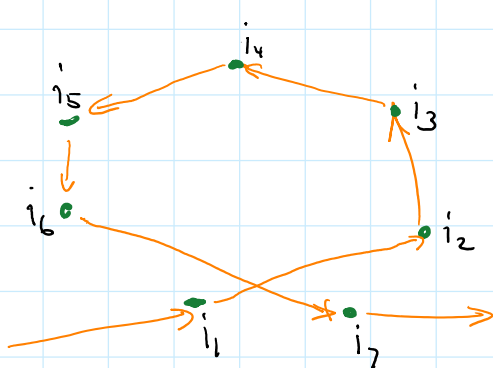
How many possible tours?

$$\frac{19 \cdot 18 \cdot 17 \cdot 16 \cdot \dots \cdot 3 \cdot 2 \cdot 1}{2} = \frac{19!}{2} = 6 \times 10^{16}$$

N cities:  $\frac{(N-1)!}{2}$  possible tours

## Simulated Annealing:

1. States: all possible tours among the N points
2. Function to minimize: total length of the tour
3. Moves between states:
  - Swap two entries in the tour.
  - Change more than two entries: swap two pairs
  - Find closest point to a random entry on the tour
  - Reverse part of the tour



tours:  $[\dots, i_1, i_2, i_3, i_4, i_5, i_6, i_7, \dots]$

$[\dots, i_1, i_6, i_5, i_4, i_3, i_2, i_7, \dots]$

reverse these indexes