Problem Statement

A 30×30 grid of squares contains 900 fleas, initially one flea per square. When a bell is rung, each flea jumps to an adjacent square at random (usually 4 possibilities, except for fleas on the edge of the grid or at the corners).

- What is the expected number of unoccupied squares after 50 rings of the bell?
- Give your answer rounded to six decimal places.

This is [Problem #213](https://projecteuler.net/problem=213) on Project Euler.

Preliminary Analysis

There are two main ways this problem can be tackled, one is actually making the fleas jump with Monte Carlo Simulation, and the other is modeling the problem as Markov Chain. This section compares the computation complexity of both, and choses that one which produces the desired result with the minimum amount of resources.

Monte Carlo Simulation

The time complexity of making n^2 fleas jump k times, and repeating the experiment m times is $O(n^2 \cdot k \cdot m)$. What we're after is the expected number of unoccupied squares, let u_l denote the number of unoccupied squared, as the result of experiment of the lth experiment, the expected value of that is:

$$\widehat{U} = \frac{1}{m}\sum_{l=1}^{m} u_l$$

The bigger m get, the more precise the estimation becomes, but how much m is supposed to be to achieve six decimal point accuracy, that is, to make $\widehat{U}$ deviate from the real value by less then $\delta = \frac{1}{2} \cdot 10^{-6}$?

The estimated value $\widehat{U}$ has uncertainty around it, which is captured by its standard error.

$$\sigma_{\widehat{U}} = \frac{\sigma_U}{\sqrt{m}}$$

Let's use a 99% confidence interval around $\widehat{U}$, which means that the real expected value $\overline{U}$ falls outside the range of $[\widehat{U} - 2.576 \cdot \sigma_{\widehat{U}}, \widehat{U} + 2.576 \cdot \sigma_{\widehat{U}}]$ with only 1% probability. The requirement to adhere to six decimal points of accuracy means that $2.576 \cdot \sigma_{\widehat{U}} < \delta$. Solving for m yields $m > 2.5 \cdot 10^{14}$. That's big.

Model as a Markov Process

This approach uses that fact that each jumping flea can be modeled as n^2 independent Markov processes.

Instead of simulating the actual jumps, this implementation …

1. Calculates the occupation density of fleas after k jumps for each flea independently
2. Invert their distribution to obtain the unoccupied squares again for each flea
3. Combine the results to calculate the expected number of free cells

See next section for a detailed example

In terms of time complexity, this can be done in $O(n^2 \cdot k)$ time.

Algorithm Design

The worked out solution seeks to lower computational complexity with proper mathematical modeling of the problem. Monte Carlo simulation is simpler to understand, but it achieves the required accuracy at a significant cost, plus executing individual experiments concurrently adds the extra complexity of coordination.

Modeling Fleas

Each flea on the $n \times n$ grid can be modeled as a Markov process that has $N = n^2$ states $S = \{1, \ldots, N\}$. The position of flea $f \in S$ at time step k is tracked by $X_f^{(k)} \in S$. The transition probabilities can be obtained from the rules on how fleas can move. For demonstration purposes, the following graph represents the model of a 3 x 3 grid.

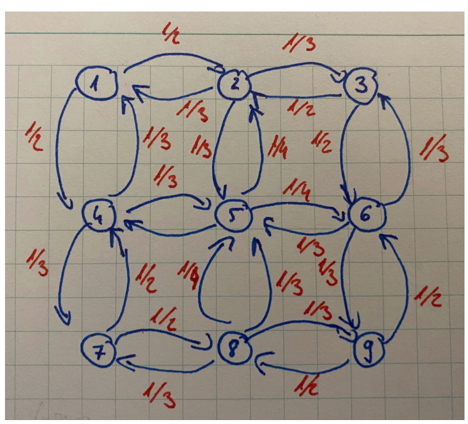

Figure 1. Markov model for a 3x3 grid

The evolution of the process of a flea's jumps starts with its initial position (state) $X_f^{(0)} \in S$ at time step $k = 0$. The starting positions are given, initially $X_f^{(0)} = f$.

Given that flea f is on square i at time step k, and the probability that it's going to move to square j at times step $k + 1$ is denoted by p_{ij}.

$$p_{ij} = P(X_f^{(k+1)} = j \mid X_f^{(k)} = i)$$

The transition probabilities between any given squares are the same for each flea, however, the probability distribution, where individual fleas might end up differ, as they have different starting positions. Let $\pi_{f,i}^{(k)}$ denote the probability distribution that flea f occupies square i at time step k. As initially every flea is on its own square, this distribution has only a single value 1 for only one square, and zero otherwise.

$$\pi_{f,i}^{(0)} = P(X_f^{(0)} = i) = \{ 1 \, if \, i \ = f \}$$

In the general case, at time step $k + 1$, the distribution $\pi_{f,i}^{(k+1)}$ evolves the following way. The probability that flea f will occupy square j depends being on square i at time k, times the probability of actually moving to j. Using the Markov property, ie. that the next step doesn't depend on squares visited before just on the current one, and accounting for all the possible ways the transition can happen, we obtain the following iterative equation.

$$\pi_{f,j}^{(k+1)} = P(X_f^{(k+1)} = j) = \sum_{i \in S} P(X_f^{(k+1)} = j \mid X_f^{(k)} = i) \cdot P(X_f^{(k)} = i) = \sum_{i \in S} p_{ij} \cdot \pi_{f,i}^{(k)}$$

For convenience, and for computation efficiency, probability densities for fleas $\pi_{f,i}^{(k)}$ can be collected to a vector $\overline{\pi_f^{(k)}} = [\pi_{f,1}^{(k)}, \ldots, \pi_{f,N}^{(k)}]$, and the transition probabilities between squares to a matrix $\overline{\overline{P}} = [p_{ij}]$. In that vector notation the previous equation can be rewritten more succinctly like this.

$$\overline{\pi_f^{(k+1)}} = \overline{\pi_f^{(k)}} \cdot \overline{\overline{P}}$$

We can also express advancing from time 0 to k at once, by computing the kth power of the transition probability matrix.

$$\overline{\pi_f^{(k)}} = \overline{\pi_f^{(0)}} \cdot \overline{\overline{P}}^k$$

$\overline{\pi_f^{(k)}}$ denotes the expected number of *occupied* squares, but what we need is the expected number of *unoccupied* squares.

$$\overline{u_f^{(k)}} = 1 - \overline{\pi_f^{(k)}}$$

$\overline{u_f^{(k)}}$ is a specific flea, and to obtain a single number, we need to multiply the probabilities, meaning that no flea occupies that square, and then sum up for all the squares.

$$U^{(k)} = \prod_{f \in S} \overline{u_f^{(k)}}$$

Example

Let’s plug-in some numbers, and unpack all of that though a concrete example.

At time 0

For the sake of simplicity, I’ll demonstrate with 2 fleas on squares 4 and 6.

0	0	0
1 (A)	0	1 (B)
0	0	0

At time 1

<table>
<tr><td></td><td>Flea A</td><td>Flea B</td></tr>
<tr><td>Occupied</td><td>$\overline{\pi^{(1)}_A}$
<table>
<tr><td>1/3</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1/3</td><td>0</td></tr>
<tr><td>1/3</td><td>0</td><td>0</td></tr>
</table></td><td>$\overline{\pi^{(1)}_B}$
<table>
<tr><td>0</td><td>0</td><td>1/3</td></tr>
<tr><td>0</td><td>1/3</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1/3</td></tr>
</table></td></tr>
<tr><td>Unoccupied</td><td>$\overline{u^{(1)}_A}$
<table>
<tr><td>2/3</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>2/3</td><td>1</td></tr>
<tr><td>2/3</td><td>1</td><td>1</td></tr>
</table></td><td>$\overline{u^{(1)}_B}$
<table>
<tr><td>1</td><td>1</td><td>2/3</td></tr>
<tr><td>1</td><td>2/3</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>2/3</td></tr>
</table></td></tr>
<tr><td>Unoccupied (neither A nor B)</td><td colspan="2">$\overline{u^{(1)}_A} \otimes \overline{u^{(1)}_B}$
<table>
<tr><td>2/3</td><td>1</td><td>2/3</td></tr>
<tr><td>1</td><td>4/9</td><td>1</td></tr>
<tr><td>2/3</td><td>1</td><td>2/3</td></tr>
</table></td></tr>
<tr><td>Total unoccupied</td><td colspan="2">$\overline{u^{(1)}_A} \cdot \overline{u^{(1)}_B} = 4 \cdot 1 + 4 \cdot \frac{2}{3} + \frac{4}{9} = \frac{64}{9} \approx 7.11$</td></tr>
</table>