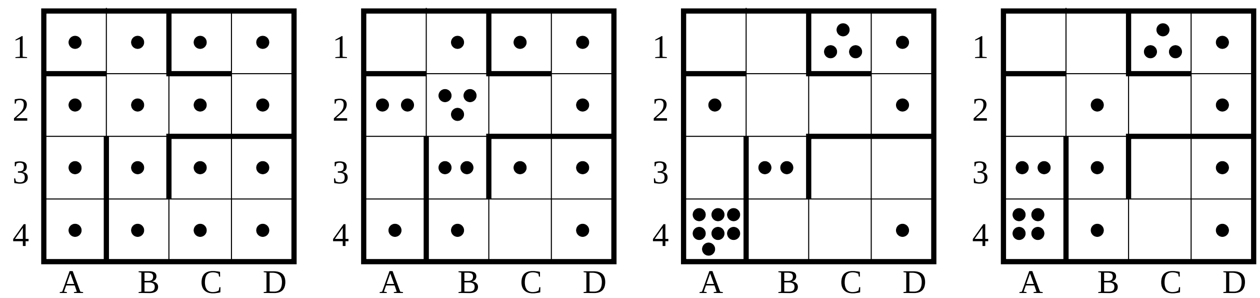
CS 6300 **Particle Filter Practice** **April 14, 2022**

You are trying to locate the ghost in the maze below, where walls are in bold. The ghost makes a clanging noise, and your acoustic sensor S can tell you the number of walls near the ghost ($W \in 0, 1, 2, 3$), with some error.



- For squares with 1 or 2 walls, the sensor S is correct 2/3rds of the time, but 1/6th of the time it overestimates by 1 or underestimates by 1.
- For squares with 0 or 3 walls, the sensor is correct 5/6ths of the time, but gets the estimate wrong by 1 in 1/6th of the time.

With equal probability, the ghost moves up, right, down or left (URDL). If there is a wall in the way, the ghost stays put. To simulate this movement, a 4-sided die maps 1-U, 2-R, 3-D and 4-L. There are 16 states (1A, 1B, etc.).

1. For convenience, states are grouped by the number of walls W surrounding each square in column 1. Fill out the CPT for $P(S|W)$. Row 1 contains the number of sensed walls. The interior of the table is to be filled with the $P(S|W)$ values.

$W\backslash S$	0	1	2	3
0	5/6	1/6	0	0
1	1/6	2/3	1/6	0
2	0	1/6	2/3	1/6
3	0	0	1/6	5/6

2. Assume you do not know where the ghost is intially. Initialize the map by placing one particle in each of the 16 squares; show this on the leftmost map. Starting with the top left, moving left to right, then top to bottom, update the location of the particles for 1 time step. Assume that the numbers rolled randomly by the die are given for each of the 16 particles as:

2, 3, 3, 1, 2, 4, 4, 3, 1, 4, 1, 2, 3, 1, 4, 2

Fill in the "New Location" row of the table below, where the particle number is 1-16 in the 1-4, A-D ordering as above. Also, show the particles in the 2nd map above.

Particle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
New Location	1B	2B	1C	1D	2B	2A	2B	2D	2A	3B	3C	3D	4A	3B	4B	4D
Re-weight	1/6	0	5/6	1/6	0	1/6	0	1/6	1/6	1/6	1/6	1/6	5/6	1/6	1/6	1/6
Normalize	16/21	0	80/21	16/21	0	16/21	0	16/21	16/21	16/21	16/21	16/21	80/21	16/21	16/21	16/21

3. After this time step, your sensor reads $S = 3$. Complete the *sensor* step of the filtering. First re-weight your particles from the previous question, then normalize them based on the sum of all the weights * 16. (I.e, if the sum of all the weights becomes 4.5, divide each particle weight by 4.5 and multiply by 16). Fill in the "Re-weight" and "Normalize" rows of the table above.

4. Now re-sample the particles with 16 new, equally weighted (1.0) particles. To do the re-sampling, use the following pseudo-code:

   for $i = 1 \rightarrow 16$ **do**
       $rand \leftarrow rand[0, 16]$
       $sum \leftarrow 0$
       for all weighted (normalized) particles j **do**
           $sum \leftarrow sum+$ weight of particle j
           if $sum > rand$ **then**
               place a new particle onto new map at this particle's location
               $break$
           end if
       end for
   end for

   Use the following randomly generated numbers for the outer for loop. Show the results in the 3rd map above and the 2nd line of the table below.

   {6.1049, 12.2483, 12.7232, 2.9900, 7.8362, 7.1294, 10.3410, 11.3498, 12.0750, 4.4164, 10.8752, 10.4816, 2.6018, 1.9040, 7.9738, 15.3559}

Particle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Location	1A	1B	1C	1D	2A	2B	2C	2D	3A	3B	3C	3D	4A	4B	4C	4D
New Location	2D	4A	4A	1C	3B	3A	4A	4A	4A	1C	4A	4A	1C	1C	3B	4D
New Location	2D	3A	4A	1C	4B	3A	3A	4A	4A	1C	4A	4A	1C	1C	3B	3D

5. Complete another time step for the distribution completed in the last step. Use these following random 'dice rolls'. Show the results on the last map and the 3rd line of the table above.

   3, 1, 3, 4, 3, 2, 1, 2, 4, 1, 4, 2, 4, 1, 2, 1