Vamshi Jandhyala

Bacteria That Cannot Leave

February 2026

A lattice-growth puzzle from the Monthly Mindbenders.[1]

Problem

A bacterium begins at the origin on the plane grid. At time 1, it divides into two bacteria that will sit just north and east of the parent, that is, at points $(0, 1)$ and $(1, 0)$. In general, at any time, any bacterium that has room to divide may do so; that is, if a bacterium sits at the point (x, y), it may be replaced by a bacterium at $(x, y+1)$ and another at $(x+1, y)$ if those points were unoccupied. Roughly, how many divisions must take place before the closed box with corners at $(0, 0)$, $(0, 3)$, $(3, 3)$ and $(3, 0)$ is clear of bacteria?

Solution

The question invites an estimate, and the honest answer is that no estimate exists. The colony grows without bound and spreads arbitrarily far from the origin, yet the box is never clear. Something always stays behind.

A quantity that never changes

Give the lattice point (x, y) in the first quadrant the weight

$$w(x, y) = \left(\tfrac{1}{2}\right)^{x+y},$$

and give a configuration of bacteria the total weight of the points it occupies. A division at (x, y) removes $w(x, y)$ and puts back

$$\left(\tfrac{1}{2}\right)^{x+y+1} + \left(\tfrac{1}{2}\right)^{x+1+y} = 2 \cdot \left(\tfrac{1}{2}\right)^{x+y+1} = \left(\tfrac{1}{2}\right)^{x+y}.$$

The two children between them carry exactly the weight of the parent. So the total weight of the colony is unchanged by every legal move, whichever bacterium divides and in whatever order. The colony starts as a single bacterium at the origin, of weight $w(0, 0) = 1$, and therefore

Proposition. At every moment, whatever the schedule of divisions, the bacteria carry total weight exactly 1.

[1] Peter Winkler, *Monthly Mindbenders*, National Museum of Mathematics, February 2026, momath.org/mindbenders, after a puzzle of Maxim Kontsevich.

How much weight the outside can hold

Now ask where that weight could possibly sit. Each lattice point holds at most one bacterium, so a set of points can hold at most its own total weight. The whole first quadrant is worth

$$\sum_{x\geq 0}\sum_{y\geq 0}\left(\tfrac{1}{2}\right)^{x+y} = \left(\sum_{x\geq 0} 2^{-x}\right)^2 = 2^2 = 4,$$

the sum factorising because $w(x,y) = 2^{-x}\cdot 2^{-y}$. The box is the sixteen points with $0 \leq x \leq 3$ and $0 \leq y \leq 3$, and the same factorisation gives its weight,

$$\left(1+\tfrac{1}{2}+\tfrac{1}{4}+\tfrac{1}{8}\right)^2 = \left(\tfrac{15}{8}\right)^2 = \frac{225}{64} \approx 3.5156.$$

Everything strictly outside the box is worth the difference,

$$4 - \frac{225}{64} = \frac{31}{64} \approx 0.4844,$$

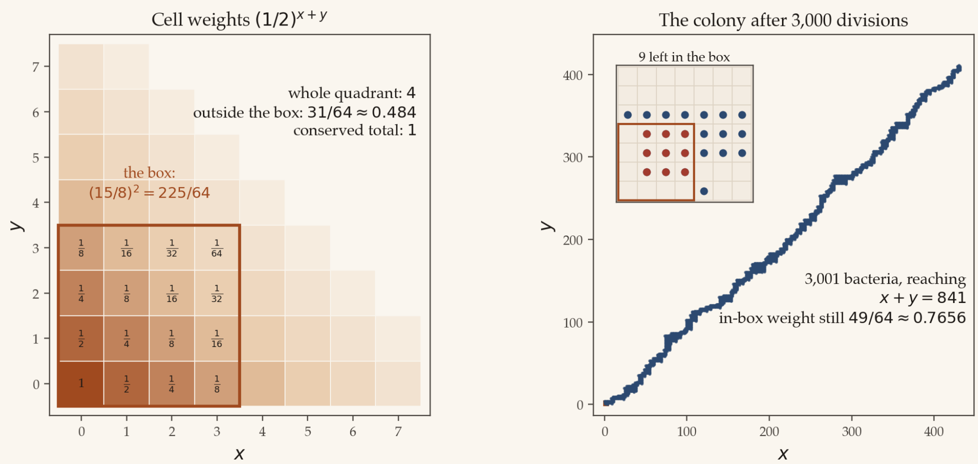
and that is a ceiling on what the outside can ever hold, reached only if every one of the infinitely many points outside the box were occupied at once. Since

$$\frac{31}{64} < 1,$$

the outside cannot accommodate the conserved weight. At least $1 - \frac{31}{64} = \frac{33}{64}$ of weight is inside the box at every moment, and weight is carried only by bacteria, so

$$\boxed{\text{the box is never clear: at least one bacterium sits inside it forever.}}$$

More than half the colony's weight, in this accounting, is permanently trapped in sixteen points near the origin.



Left: the weight $(1/2)^{x+y}$ *on each cell. The box alone is worth* 225/64 *of the quadrant's* 4*, leaving* 31/64 *outside. Right: a colony after* 3,000 *divisions, reaching* $x + y = 841$ *while nine bacteria (inset, in red) remain in the box.*

Why the intuition fails

The pull of the question is the sense that a growing colony must eventually vacate any fixed patch, since it keeps moving north and east. It does grow: after n divisions there are $n+1$ bacteria, and the front runs off to infinity. The weight argument says that the growth is the wrong thing to watch. Motion away from the origin is exactly what costs weight, and a fixed budget of 1 buys only so much of it. The bacteria that have travelled far carry almost nothing, so they cannot relieve the ones near the origin of their share.

Nothing in the argument is probabilistic, and nothing depends on the order of divisions, so the bound holds even against an adversary who picks each move with the express aim of clearing the box, and holds in the limit too, since passing to a limit can only lose weight from the outside region.

What the argument reaches

The proof has two halves that vary independently. Conservation comes from the identity $2 \cdot \frac{1}{2} = 1$, specific to this splitting rule. The capacity count applies to any region at all.

Call a finite region R of the quadrant *unclearable* if the bacteria can never all be outside it. The count above proves this whenever the complement of R is worth less than 1, that is, whenever

$$w(R) > 4 - 1 = 3.$$

For the square box $\{0, \dots, n\}^2$ the weight is

$$(2 - 2^{-n})^2,$$

which is $\frac{9}{4}$ at $n = 1$ and $\frac{49}{16} = 3.0625$ at $n = 2$, and increases to 4. So every square box anchored at the origin with $n \geq 2$ is unclearable, and the puzzle's box, at $n = 3$, is not the smallest one with the property. The three-by-three square $\{0, 1, 2\}^2$ already traps a bacterium forever, with the tighter margin $1 - \frac{15}{16} = \frac{1}{16}$.

Triangles behave the same way. The staircase $\{x + y \leq k\}$ has $k+1$ points on its outer diagonal and weight

$$\sum_{j=0}^{k} \frac{j+1}{2^j},$$

which is $\frac{11}{4}$ at $k = 2$ and $\frac{13}{4}$ at $k = 3$. So some bacterium always sits within $x + y \leq 3$ of the origin, however long the process runs. At $k = 2$ the weight falls below 3 and the argument goes quiet, which is the usual position of a capacity bound: it is sufficient, not necessary, and where it stops speaking the question stays open.

The shape of the reasoning travels further than the puzzle. A move that leaves some quantity exactly invariant, paired with a region whose capacity for that quantity is too small, forbids an outcome without any accounting of how the process runs. Conway's soldiers is the same manoeuvre, with a weight that decays as φ^{-d} in the distance d from the target square, the golden ratio being forced by $\varphi^{-2} + \varphi^{-1} = 1$, the identity that makes a jump conserve weight. The choice of $\frac{1}{2}$ here is forced in the same way: it is the number that makes two children weigh what one parent weighed.

Python code

The arithmetic above is hard to get wrong, so the code earns its place elsewhere: it plays the game. It keeps the occupied set, picks a bacterium with room, divides it, and does so under two schedules, one random and one that always divides the outermost bacterium still in the box in an effort to empty it. Both run for two hundred thousand divisions.

```
import heapq, random

W   = lambda x, y: 0.5**(x + y)         # the weight of the cell (x, y)
BOX = [(x, y) for x in range(4) for y in range(4)]
box = sum(W(*c) for c in BOX)
print(f"quadrant {4.0:.4f} box {box:.4f} outside {4-box:.4f}"
      f" room outside for weight 1? {4-box >= 1}")

def run(order, steps=200_000):          # `order` fixes the schedule
    occ, heap, tick = {(0, 0)}, [], 0
    free = lambda c: (c in occ and (c[0]+1, c[1]) not in occ
                        and (c[0], c[1]+1) not in occ)
    def offer(c):                       # a cell that may now divide
        nonlocal tick; tick += 1
        if free(c): heapq.heappush(heap, (order(c), tick, c))
    offer((0, 0))
    for n in range(1, steps + 1):
        while True:                     # drop cells walled in since
            c = heapq.heappop(heap)[2]
            if free(c): break
        x, y = c
        occ.remove(c); occ |= {(x, y+1), (x+1, y)}
        for a, b in ((x, y), (x, y+1), (x+1, y)):
            for e in ((a, b), (a-1, b), (a, b-1)): offer(e)
        if n in (1_000, 10_000, 100_000, 200_000):
            ib = [c for c in BOX if c in occ]
            print(f" {n:>7,}: {len(occ):>7,} bacteria, {len(ib)} in box,"
                  f" in-box weight {sum(W(*c) for c in ib):.4f},"
                  f" total {sum(W(*c) for c in occ):.6f}")

random.seed(2026)
print("a random schedule")
run(lambda c: random.random())
print("a schedule that empties the box as fast as it can")
run(lambda c: (0, -(c[0]+c[1])) if max(c) < 4 else (1, c[0]+c[1]))
# quadrant 4.0000 box 3.5156 outside 0.4844 room outside for weight 1? False
# a random schedule
#    1,000:  1,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#   10,000: 10,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#  100,000: 100,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#  200,000: 200,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
# a schedule that empties the box as fast as it can
#    1,000:  1,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#   10,000: 10,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#  100,000: 100,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
#  200,000: 200,001 bacteria, 9 in box, in-box weight 0.7656, total 1.000000
```

Both schedules settle on the same nine survivors, the square $\{1,2,3\}^2$ of weight $\frac{49}{64}$, each walled in by its northern and eastern neighbour. That floor sits above the guaranteed $\frac{33}{64}$, as one expects of a bound that assumes every outside point occupied at once.