Vamshi Jandhyala

The Corner No Peg Can Reach

✠

August 2026

A parity puzzle from the Monthly Mindbenders.[1]

Problem

Three pegs stand at three corners of a square in the plane. At any time you may jump one peg over another, landing it the same distance beyond, on the far side. The peg jumped over stays where it is. Can a peg ever be brought to the fourth corner of the square?

Solution

No. The proof is a single invariant, but it is worth first clearing away the square's size and angle, since neither can matter.

A jump sends a peg at A over a peg at B to the point $2B - A$, the reflection of A through B. Any map of the plane that preserves straight lines and midpoints, a stretch, a rotation, a shear or a shift, carries the point $2B - A$ to $2f(B) - f(A)$, so it turns legal jumps into legal jumps. One such map carries the square onto the unit square, with the pegs at $(0,0)$, $(1,0)$ and $(0,1)$ and the empty corner at $(1,1)$. From here on the pegs sit on points with integer coordinates, and every jump keeps them there.

Now look at each coordinate modulo 2, whether it is even or odd. The jumping peg lands at

$$2B - A \;\equiv\; -A \;\equiv\; A \pmod 2,$$

coordinate by coordinate, since $2B$ is even in both coordinates. So a jump never changes the parity class of the peg that moves, and the peg jumped over does not move at all. Every peg keeps its parity class for the whole game.

The three pegs start in the classes $(\text{even}, \text{even})$, $(\text{odd}, \text{even})$ and $(\text{even}, \text{odd})$. The fourth corner, $(1,1)$, is in the class (odd, odd), which no peg has ever belonged to and none ever will.

No peg can reach the fourth corner.

The argument proves rather more than was asked. The peg that starts at the origin can only ever stand on points with both coordinates even, so it can never reach the

[1] Peter Winkler, *Monthly Mindbenders*, National Museum of Mathematics, August 2026, conveyed by Paul Zeitz, momath.org/mindbenders.

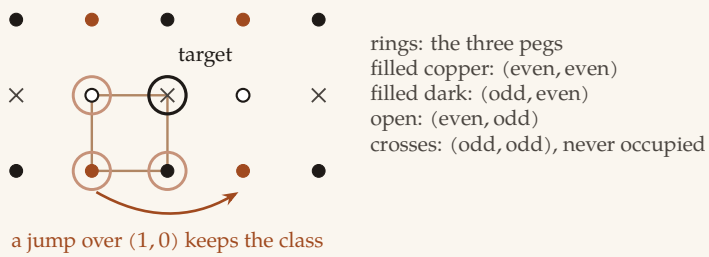

Figure 1: *Lattice points marked by parity class. Each peg can only ever visit points of its own class, and the fourth corner belongs to the one class that holds no peg.*

corner $(1, 0)$ either, even though another peg began there. The pegs keep one each of three of the four classes, permanently, and the square's fourth corner lies in the fourth.

None of this means the pegs are confined in any other way. They wander freely over their own classes: a single jump of the origin peg over $(1, 0)$ lands it at $(2, 0)$, and a few jumps reach points in every direction. The only thing ruled out is the class nobody started in.

The computation

A search of every configuration reachable from the start, with all pegs kept inside a window of twelve units either way, checks the conclusion without using the invariant. It then checks that the search could have found a violation, by rerunning it with a deliberately broken rule that lands the peg one step off.

```
from collections import deque

def reach(start, jump, bound=12, cap=400_000):
    seen, q, pts = {start}, deque([start]), set(start)
    while q and len(seen) < cap:
        cfg = q.popleft()
        for i in range(3):
            for j in range(3):
                if i != j:
                    na = jump(cfg[i], cfg[j])
                    if max(map(abs, na)) <= bound:
                        new = tuple(sorted(cfg[:i] + (na,) + cfg[i+1:]))
                        if new not in seen:
                            seen.add(new); q.append(new); pts.add(na)
    return seen, pts

start = ((0, 0), (0, 1), (1, 0))
cfgs, pts = reach(start, lambda a, b: (2*b[0]-a[0], 2*b[1]-a[1]))
print(len(cfgs), len(pts), (1, 1) in pts) # 61868 481 False
```

The search visits 61,868 configurations and 481 distinct peg positions, never the point $(1, 1)$, and in every configuration the three pegs occupy three different parity classes. The broken rule, which adds 1 to the landing point's first coordinate, puts a peg into the forbidden class (odd, odd) at once, so the check is capable of failing.