1 The Problem

You're on a two-dimensional grid $\mathbb{Z} \times \mathbb{Z}$ and have to find a way to get to one coordinate $(x, y) \in \mathbb{Z} \times \mathbb{Z}$. You start at $(0, 0)$.

In your i-th step you move either $\underbrace{(+i, 0)}_{=:E}$, $\underbrace{(-i, 0)}_{=:W}$, $\underbrace{(0, +i)}_{=:N}$ or $\underbrace{(0, -i)}_{=:S}$.

2 The algorithm

Algorithm 1 Algorithm to calculate the minimum amount of steps

function CALCULATESTEPS($x \in \mathbb{Z}$, $y \in \mathbb{Z}$)
  $s \leftarrow 0$
  $dist \leftarrow |x| + |y|$

  while $\overbrace{\frac{s^2 + s}{2} < dist}^{\text{condition 1}}$ or $\overbrace{\frac{s^2 + s}{2} \not\equiv dist \pmod 2}^{\text{condition 2}}$ **do**
    $s \leftarrow s + 1$
  end while

  return s
end function

Algorithm 2 Algorithm to solve the pogo problem

```
function solvePogo(x ∈ ℤ, y ∈ ℤ)
    s_min ← calculateSteps(x, y)

    solution ← ε
    for i in s_min, ..., 1 do
        if |x| > |y| then
            if x > 0 then
                solution ← solution + E
                x ← x − i
            else
                solution ← solution + W
                x ← x + i
            end if
        else
            if y > 0 then
                solution ← solution + N
                y ← y − i
            else
                solution ← solution + S
                y ← y + i
            end if
        end if
    end for

    return solution
end function
```

3 Correctness

3.1 calculateSteps

Let $x, y \in \mathbb{Z}$ and $s := \textsc{calculateSteps}(x, y)$.

Let $s_{\min}$ be the minimum amount of necessary steps to get from $(0, 0)$ to (x, y) when you move i units in your i'th step.

Theorem: $s = s_{\min}$

It's enough to proof $s \geq s_{\min}$ and $s \leq s_{\min}$.

Theorem: $s \leq s_{\min}$ (we don't make too many steps)

Proof:

We have to get from $(0, 0)$ to (x, y). As we may only move in taxicab geometry we have to use the taxicab distance measure d_1:

$$d_1(p, q) := \sum_{i=1}^{2} |p_i - q_i|$$

So in our scenario:

$$d_1((0, 0), (x, y)) = |x| + |y|$$

This means we have to move at least $|x|+|y|$ units to get from $(0,0)$ to (x,y). As we move i units in the i'th step, we have to solve the following equations for $s_{\min 1}$:

$$\sum_{i=1}^{s_{\min 1}} i \geq |x|+|y| \qquad \text{and} \qquad |x|+|y| > \sum_{i=1}^{s_{\min 1}-1} i \tag{1}$$

$$\frac{s_{\min 1}^2 + s_{\min 1}}{2} \geq |x|+|y| \qquad > \sum_{i=1}^{s_{\min 1}-1} i \tag{2}$$

This is what algorithm 1 check with condition 1. As the algorithm increases s only by one in each loop, it makes sure that $\sum_{i=1}^{s_{\min 1}-1} i$ is bigger than $|x|+|y|$.

You can undo moves by going back. But this will always make an even number undone. When you go $(+i,0)$ and later $(-j,0)$ it is the same as if you've been going $(i-j,0)$. So $2 \cdot i$ steps got undone. But $2 \cdot i$ is an even number. You will never be able to undo an odd number of moved units. This means, the parity of the minimum number of units you would have to move if you would move one unit per step has to be the same as the parity of the moves you actually do. This is exactly what condition two makes sure.

So we need at least s steps $\Rightarrow s \leq s_{\min}\square$

Theorem: $s \geq s_{\min}$ (we make enough steps)

Proof:

TODO

3.2 solvePogo

Theorem: solvePogo(x,y) returns a valid, minimal sequence of steps to get from $(0,0)$ to (x,y)

Proof:

TODO