--- Day 20: Particle Swarm ---

Suddenly, the GPU contacts you, asking for help. Someone has asked it to simulate too many particles, and it won't be able to finish them all in time to render the next frame at this rate.

It transmits to you a buffer (your puzzle input) listing each particle in order (starting with particle `0`, then particle `1`, particle `2`, and so on). For each particle, it provides the `X`, `Y`, and `Z` coordinates for the particle's position (`p`), velocity (`v`), and acceleration (`a`), each in the format `<X,Y,Z>`.

Each tick, all particles are updated simultaneously. A particle's properties are updated in the following order:

- Increase the `X` velocity by the `X` acceleration.
- Increase the `Y` velocity by the `Y` acceleration.
- Increase the `Z` velocity by the `Z` acceleration.
- Increase the `X` position by the `X` velocity.
- Increase the `Y` position by the `Y` velocity.
- Increase the `Z` position by the `Z` velocity.

Because of seemingly tenuous rationale involving z-buffering, the GPU would like to know which particle will stay closest to position `<0,0,0>` in the long term. Measure this using the Manhattan distance, which in this situation is simply the sum of the absolute values of a particle's `X`, `Y`, and `Z` position.

For example, suppose you are only given two particles, both of which stay entirely on the X-axis (for simplicity). Drawing the current states of particles `0` and `1` (in that order) with an adjacent a number line and diagram of current `X` positions (marked in parenthesis), the following would take place:

```
p=< 3,0,0>, v=< 2,0,0>, a=<-1,0,0>    -4 -3 -2 -1  0  1  2  3  4
p=< 4,0,0>, v=< 0,0,0>, a=<-2,0,0>                         (0)(1)

p=< 4,0,0>, v=< 1,0,0>, a=<-1,0,0>    -4 -3 -2 -1  0  1  2  3  4
p=< 2,0,0>, v=<-2,0,0>, a=<-2,0,0>                      (1)   (0)

p=< 4,0,0>, v=< 0,0,0>, a=<-1,0,0>    -4 -3 -2 -1  0  1  2  3  4
p=<-2,0,0>, v=<-4,0,0>, a=<-2,0,0>          (1)               (0)

p=< 3,0,0>, v=<-1,0,0>, a=<-1,0,0>    -4 -3 -2 -1  0  1  2  3  4
p=<-8,0,0>, v=<-6,0,0>, a=<-2,0,0>                         (0)
```

At this point, particle `1` will never be closer to `<0,0,0>` than particle `0`, and so, in the long run, particle `0` will stay closest.

Which particle will stay closest to position `<0,0,0>` in the long term?

Your puzzle answer was `119`.

--- Part Two ---

To simplify the problem further, the GPU would like to remove any particles that collide. Particles collide if their positions ever exactly match. Because particles are updated simultaneously, more than two particles can collide at the same time and place. Once particles collide, they are removed and cannot collide with anything else after that tick.

For example:

```
p=<-6,0,0>, v=< 3,0,0>, a=< 0,0,0>
p=<-4,0,0>, v=< 2,0,0>, a=< 0,0,0>    -6 -5 -4 -3 -2 -1  0  1  2  3
p=<-2,0,0>, v=< 1,0,0>, a=< 0,0,0>    (0)   (1)   (2)            (3)
p=< 3,0,0>, v=<-1,0,0>, a=< 0,0,0>

p=<-3,0,0>, v=< 3,0,0>, a=< 0,0,0>
p=<-2,0,0>, v=< 2,0,0>, a=< 0,0,0>    -6 -5 -4 -3 -2 -1  0  1  2  3
p=<-1,0,0>, v=< 1,0,0>, a=< 0,0,0>             (0)(1)(2)      (3)
p=< 2,0,0>, v=<-1,0,0>, a=< 0,0,0>

p=< 0,0,0>, v=< 3,0,0>, a=< 0,0,0>
p=< 0,0,0>, v=< 2,0,0>, a=< 0,0,0>    -6 -5 -4 -3 -2 -1  0  1  2  3
p=< 0,0,0>, v=< 1,0,0>, a=< 0,0,0>                       X (3)
p=< 1,0,0>, v=<-1,0,0>, a=< 0,0,0>

------destroyed by collision------
------destroyed by collision------    -6 -5 -4 -3 -2 -1  0  1  2  3
------destroyed by collision------                      (3)
p=< 0,0,0>, v=<-1,0,0>, a=< 0,0,0>
```

In this example, particles `0`, `1`, and `2` are simultaneously destroyed at the time and place marked `X`. On the next tick, particle `3` passes through unharmed.

How many particles are left after all collisions are resolved?

Your puzzle answer was `471`.

Both parts of this puzzle are complete! They provide two gold stars: **

At this point, you should return to your advent calendar and try another puzzle.

If you still want to see it, you can get your puzzle input.

You can also [Share] this puzzle.