Advent of Code 2021 Day 17

AlcatrazEscapee

December 2021

1 Introduction

The goal is to enumerate all possible trajectories that land in the target region, in a finite manner. A trajectory is defined as an initial velocity, $(u, v) \in \mathbb{Z}^2$. As the target region itself is bounded, we can reduce this problem to one of enumerating all possible trajectories that at some $n \in \mathbb{Z}_{\geq 0}$, we have $(x(n), y(n)) = (x_T, y_T)$.

1.1 Vertical Position

Consider then an arbitrary trajectory which at some n will reach the point (x_T, y_T) in the target region. First, we examine the y velocity of such a trajectory. The velocity is governed by the following equations as described in the problem statement:

$$
\begin{aligned}
v_y(n) &= v - n \\
y(n) &= \sum_{j=1}^{n} v_y(j-1) && \text{if}\, n > 0 \\
&= 0 && \text{if}\, n = 0
\end{aligned}
$$

We can then solve the equation for $y(n)$:

$$
\begin{aligned}
y(n) &= \sum_{j=1}^{n} v_y(j-1) \\
&= \sum_{j=1}^{n} (v - j + 1) \\
&= nv + n - \sum_{j=1}^{n} j \\
&= nv + n - \frac{1}{2}n(n+1)
\end{aligned}
$$

Thus, for a trajectory with y velocity v, that at some n reaches the position y_T, we have that:

$$y_T = nv + n - \frac{1}{2}n(n+1)$$

The above equation can be re-arranged into the following form:

$$2y_T = n(1 + 2v + n)$$

Which introduces a useful constraint. As $1 + 2v + n$ is an integer, we must have that n divides $2y_T$. This then allows us to enumerate possible values of n in a straightforward fashion, by checking if $2y_T = 0 \bmod n$. In addition, once a value of n is found in this manner, it gives an explicit formula for v:

$$\begin{aligned} v &= \frac{1}{n}y_T - 1 + \frac{1}{2}(n+1) \\ &= \frac{2y_T - 2n + n(n+1)}{2n} \\ &= \frac{2y_T - n + n^2}{2n} \end{aligned}$$

Note that as v is restricted to be an integer, this can then be implemented by checking if $2y_T - n + n^2 = 0 \bmod 2n$, and if true, a possible pair of n and v has been identified.

This effectively reduces the problem of finding valid y velocities, given a target y value, to one of iterating over the range $n \in \{1, \ldots, 2|y_T|\}$, for all y_T.

1.2 Horizontal Position

The second step, given an existing pair of n and v, is to enumerate possible values of u that satisfy an arbitrary x_T. As before, we can examine the equations governing the position and the velocity in the x direction. Without loss of generality, we will assume that $x_T > 0$, which implies that $u > 0$. For regions where both values are negative, the same results apply, and for regions where $x_T = 0$, the only possible value of u is zero.

$$\begin{aligned} v_x(n) &= \max\{0, u - n\} \\ x(n) &= \sum_{j=1}^{u} v_x(j-1) && \text{if}\, n \geq u \\ &= \sum_{j=1}^{n} v_x(j-1) && \text{if}\, 1 \leq n < u \\ &= 0 && \text{if}\, n = 0 \end{aligned}$$

Note that the above equations imply there are finite values that $x(n)$ can take, for values of $n \in \{0, 1, \ldots u\}$. However, note that if we reach a target x position at $n = u$, there are infinitely many n which will also have that same target x position. We can consider both cases separately. For a fixed n, if we have that $u \leq n$:

$$
\begin{aligned}
x(n) &= \sum_{j=1}^{u} v(j-1) \\
&= \sum_{j=1}^{u} (u-j+1) \\
&= u^2 + u - \sum_{j=1}^{u} j \\
&= u^2 + u - \frac{1}{2}u(u+1)
\end{aligned}
$$

In particular, if we let $x_T = x(n)$:

$$
\begin{aligned}
x_T &= x(n) \\
x_T &= u^2 + u - \frac{1}{2}u(u+1) \\
2x_T &= u(u+1)
\end{aligned}
$$

Since we assumed without loss of generality that $u > 0$, and we have by prior assumption that $u \leq n$, we can simply enumerate all values of u and check for those that satisfy the above condition. However, we can further reduce this problem by considering useful bounds based on $K = \sqrt{2x_T}$.

$$
\begin{aligned}
u(u+1) &= u^2 + u \\
&> u^2 \\
&> K^2 \quad \text{for } K < u \\
&= 2x_T
\end{aligned}
$$

Thus from the above, we can assert that we must have $u \leq K$. We can similarly bound below:

$$
\begin{aligned}
u(u+1) &= u^2 + u \\
&< (u+1)^2 \\
&< K^2 \quad \text{for } u+1 < K \\
&= 2x_T
\end{aligned}
$$

Thus we must also have that $u \geq K - 1$. Adjusting for integer values, this means we only have to check a fixed number of values of u, at the cost of one square root operation.

In the second case, we consider values of u where $u > n$. We can then solve the equation for the x position in a similar fashion to the one for the y position:

$$
\begin{aligned}
x(n) &= \sum_{j=1}^{n} v(j-1) \\
&= \sum_{j=1}^{n} (u-j+1) \\
&= nu + n - \sum_{j=1}^{n} j \\
&= nu + n - \frac{1}{2}n(n+1)
\end{aligned}
$$

And as before, assuming at some n, the trajectory reaches a point x_T, we find that:

$$
\begin{aligned}
x_T &= x(n) \\
x_T &= nu + n - \frac{1}{2}n(n+1) \\
2x_T &= n(1+2u+n)
\end{aligned}
$$

Which gives us the final constraint. As $1+2u+n$ is an integer, we must have that n divides $2x_T$. Sine n is already determined by the equation for the y velocity, we can then check if $2x_T = 0 \bmod n$. And as before, we can construct an explicit formula for u:

$$
\begin{aligned}
u &= \frac{1}{n}x_T - 1 + \frac{1}{2}(n+1) \\
&= \frac{2x_T - 2n + n(n+1)}{2n} \\
&= \frac{2x_T - n + n^2}{2n}
\end{aligned}
$$

Which derives the same additional condition: As u is restricted to be an integer, we can then check if $2x_T - n + n^2 = 0 \bmod 2n$, and if true, compute the value of u.

Finally, we need to analytically compute the maximum y value reached during all possible trajectories. As we can now enumerate all possible trajectories in an explicit fashion, the question becomes one of finding the maximum possible y value of a given trajectory with initial velocity (u, v).

There are two cases: if $u \leq 0$, we trivially know that $y_{max} = 0$ for the rest of the trajectory. If $u > 0$, we can apply a principle from physics which says that the maximum height will be attained when the y velocity is at a minimum, or zero. Using the explicit equations for y velocity and position above, we can derive:

$$
\begin{aligned}
v_y(n) &= v - n = 0 \\
\therefore\ n &= v \\
y(n) &= nv + n - \frac{1}{2}n(n+1) \\
y_{max} &= y(v) \\
&= v^2 + v - \frac{1}{2}v(v+1) \\
&= \frac{1}{2}v(v+1)
\end{aligned}
$$

1.3 Summary

In summary, we can iterate first, through every y_T within the target region, and every $n \in \{1, \ldots 2|y_T|\}$ which satisfies the equality $2y_T = 0 \bmod n$ and $2y_T - n + n^2 = 0 \bmod 2n$.

With this, a pairing of v and n have been found, we can then iterate through every x_T in the target region, and check the two cases above. In the case where $u \leq n$, given $K = \sqrt{2x_T}$, we check the values of $u \in \{\lfloor K \rfloor - 1, \ldots, \lceil K \rceil\}$ for any which satisfy $u(u+1) = 2x_T$. In the second case, we can check any if the current value of n satisfies the equality $2x_T = 0 \bmod n$ and $2x_T - n + n^2 = 0 \bmod 2n$. This is a sufficient condition in order to enumerate all possible pairings of (u, v) that eventually end up in the target area.

Finally, in order to calculate the maximum attained velocity we simply iterate through all found possible v values and calculate the maximum of $\frac{1}{2}v(v+1)$ if $v > 0$.

2 Implementation

An implementation in Python using the above techniques is included below. This implementation assumes the min and max bounds have already been set, and also assumes without loss of generality that the minimum x bound is positive.

```python
velocities = set()
for yt in range(min_y, 1 + max_y):
  for n in range(1, 1 + 2 * abs(yt)):
    if (2 * yt) % n == 0 and (v0 := 2 * yt - n + n * n) % (2 * n) == 0:
      v = v0 // (2 * n)
      for xt in range(min_x, 1 + max_x):
        k = sqrt(2 * xt)
        for u in range(floor(k), 1 + ceil(k)):
          if u <= n and u * (u + 1) == 2 * xt:
            velocities.add((u, v))
        if (2 * xt) % n == 0 and (u0 := 2 * xt - n + n * n) % (2 * n) == 0:
          u = u0 // (2 * n)
          if u > n:
            velocities.add((u, v))

print('Part 1:', max(v * (v + 1) // 2 for u, v in velocities))
print('Part 2:', len(velocities))
```